Universidade de Aveiro Departamento de Ambiente e Ordenamento
2018

António Pedro Tão Nunes Bertelo

Metodologia Inversa para Calibração de Modelos Numéricos de Sistemas de Abastecimento de Água

Inverse Methodology for Calibration of Numerical Models of Water Supply Systems

Universidade de Aveiro Departamento de Ambiente e Ordenamento
2018

António Pedro Tão Nunes Bertelo

Metodologia Inversa para Calibração de Modelos Numéricos de Sistemas de Abastecimento de Água

Inverse Methodology for Calibration of Numerical Models of Water Supply Systems

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Sistemas Energéticos e Sustentáveis, realizada sob a orientação científica do Doutor António Gil d'Orey de Andrade Campos, Professor Auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro e do co-orientador Mestre Miguel Oliveira, professor assistente convidado do departamento de economia, gestão, engenharia industrial e turismo.

Dedico a toda a minha família e amigos.

o júri

presidente

Prof.ª Doutora Mara Teresa da Silva Madaleno
Professora Auxiliar da Universidade de Aveiro

Prof. Doutor João Alexandre Dias de Oliveira
Professor Auxiliar da Universidade de Aveiro

Prof. Doutor António Gil D'Orey de Andrade Campos
Professor Auxiliar da Universidade de Aveiro (orientador)

agradecimentos

A todos que contribuíram para a realização desta dissertação, direta ou indiretamente.

palavras-chave redes de abastecimento de água, calibração, modelo inverso, fugas nodais, otimização

resumo

Uma vez que as despesas das empresas de gestão de água abrangem os gastos das operações do sistema de distribuição de água, é então necessário adequar o sistema à variabilidade do consumo de água dos consumidores atendendo ao custo do bombeamento. A calibração de consumos nodais de um sistema de distribuição de água é um processo de aproximação de valores previstos calculados através de um modelo hidráulico para com os valores observados. Para este caso é considerado um sistema de abastecimento em baixa. Foram analisados dois métodos incluídos no processo de calibração dos consumos nodais: i) método inverso clássico e o ii) método do modelo inverso.
O método do modelo inverso incorpora o método de Gradiente desenvolvido por Todini. Este método é conhecido pela resolução de um sistema de equações de balanço de massa e energia. O modelo inverso apresenta-se como o mais eficiente e é constituído pelo processo de separação de variáveis conhecidas e desconhecidas das equações de continuidade e energia da rede de abastecimento de água. Os consumos nodais com características similares são agregados para tornar o modelo determinado. O método de Gauss-Newton é então aplicado para resolver o modelo. Ambos os métodos foram analisados através de um caso de estudo constituído por uma rede hidráulica simples, com o propósito de simular condições iniciais segundo um regime permanente e transiente.

keywords water supply networks, calibration, inverse model, nodal demand, optimization

abstract

Since the costs of the water management companies cover the costs of the operations of the water distribution system, it is, therefore, necessary to adapt the water consumption of the consumers with the best price negotiated by the companies. The calibration of nodal demands of a water distribution system is a process of approximation of the predicted values calculated through a hydraulic model to the observed values. Two methods for calibration of nodal demands were analyzed: i) Classical Inverse method and ii) Inverse model method.
The Inverse model method incorporates the Gradient method developed by Todini. This method is known for solving the system of mass and energy balance equations. The inverse model presents itself as more efficient and is constituted by the separation of known and unknown variables from the continuity and energy equations of the water supply network. Nodal demands with similar characteristics are aggregated to make the model determined. The Gauss-Newton method is then applied to solve the model. Both methods were analyzed through a case study of a simple hydraulic network for a transient and permanent regime.

Contents

1. Introduction

1.1 Approaches to the water sector

The water, undoubtedly as part of our body constitution, had over the years a preponderance in the growth and development of the human species, whereas the greatest expansion dates to the 20th century, with the population almost quadruple. This strong demographic evolution has grown the dependence of the human being on water and energy. Consequently, these two driving forces have been tremendously exploited in such a way that the expenses inherent to the transportation and distribution of water have been reviewed and analyzed in order to build a more efficient supply network than the current [1].

1.2 Sector problems

The main concern of companies of management supply networks, most of them municipal companies, is providing water with effectiveness and with a standard quality, trying as much as possible reduce the costs involved in pumping water from its source up to the client consumer.

It is estimated that the world cost associated with water pumping is about 12 billion euros. This value assumes a considerable dimension and at the same time raises a lot of questions in terms of energy resources optimization and in the focus of the available technology [2]. To that extent there is a countless of aspects relevant to distribution and supply system operation, where it can be integrated the leakage, pressure losses, conducts ruptures and mainly a bad daily management of pumps, reflecting many inefficiencies at the energy level, being this point the focus of the present work.

1.3 Solution presented

The need for a more effective strategy for this kind of supply networks brings benefits to saving a vital and scarce resource and to reduce the population's monthly bill.

Once the expenses linked by municipal companies cover the costs for the operations of water distribution system, then the need arises to adapt the system to the variability of consumer's water demand in relation to the price to pay for it. This optimization process allows companies to be informed of the best time to pumping water, and in the ideal case, this support system is based on the lower rate of the energy market, always with the guarantee that the minimum volume of water required in the reservoir is assured. At this point, the simulation process is run, making the connection between the hydraulic model and the variables unknown of the network behavior.

Similarly, it is possible to know the consumption pattern of a given supply network, and thereby predict the energy output according to the tariff negotiated with the power companies.

The simulation process is only useful and capable of representing the distribution water network if the model is calibrated.

1.4 Difficulties and challenges of calibration

The calibration of water supply networks has been over the last few decades an experimental connotation based on the investigation of field professionals. This makes the information available about actual and real data be compared to solve the high uncertainty about the measurements collected. This context has been set as a challenge to improve the network constraints surpassing the barriers previously existent [3].

The solution of this calibration problem is based on the minimization of an objective function that defines the difference between the real data and simulated observations of hydraulic properties, such as flows and pressures. Many of the hydraulic model constituents have unknown parameters and input data that can compromise the accuracy of the results, as the case of material wear or valve damage beyond leakage or ruptures.

The calibration follows a methodology that requires several measurements, including flow rates, pressures, roughness, brings a difficulty in filtering the essential points that best personify the basic structure of hydraulic system model [4].

1.5 Objectives

The main objective of this work is to find out an efficient calibration method for a hydraulic model of a water supply network and compare it to the classical try-and-error methodology. It is intended that the developed method would be more efficient than the current methods that solve the problem by minimizing the direct problem through optimization.

The resolution of the problem has two different methods, the first relates to classical roots respecting an implicit approach using optimization. The second is related with the separation of the known and unknown variables to solve the undetermined model. The purpose of calibrating nodal demands by the two approaches is used for validating a Case study. The numerical algorithm of both methods is inserted in the Case Study. The main goal of the Case study is to show that both methods can solve the problem in a feasible way.

1.6 Guidelines

This thesis is divided into six chapters. The first chapter presents a brief introduction to the subject. The second chapter, which is divided into several sections, is dedicated to the detailed characterization of the subject as well as define the main concepts inherent to the scope of the developed theme. The third chapter presents the numerical calibration methodology.

The fourth chapter contains the results and validation of the proposed methodology where the case study is applied.

The last chapter of this dissertation ends with the final considerations, followed by attaching the several files in the Appendix A and B. Appendix A provides an overview about the Steady-state operations and results of the Case study created. Appendix B provides also the results and operations conditions but related with the Transient regime.

2. State of the art review

2.1 Modeling hydraulic water supply networks

During the calibration process of a model, is implied the context of modeling, which works with a mathematical model to simulate behaviors of the network. A great capacity for identification and resolution of problems can bring an appropriate model for a specific network and that kind of process could be the key to successful water distribution system. In this type of systems, the mathematical model should simulate the real network behavior under specific conditions during operation. Regardless of complexity, mathematical models try to reflect a variety of data that have a future preponderance in the knowledge and analysis of situations resulting from network imperfections [5].

It is expected that hydraulic modeling will bring large advantages, in addition, on monitoring the quality of water supply in case of leakage and the control of those same losses especially in terms of energy and financial operating costs. In that situation could be considered the optimization between the minimum volume of water in the reservoir and the timeline (hours) of pumping as an interesting solution to improving the system.

The integration of hydraulic simulation in real time has been a clear emphasis on the knowledge of the network patterns. The data collected can be viewed online in real time and thus having a strong and clear adaptation with auxiliary forecast systems. In this way, the flow rates and pressures will be stored in "cloud", where through the information captured by the sensors spread across multiple nodes, they will be able to forecast and preview the network status with a 24-hour advance [6]. The whole process of hydraulic modeling is not limited to the characterization of the model, however, the greatest challenge is the calibration and the validation of the results recorded [6].

2.1.1 Hydraulics models and methodologies

In order to match the pumping periods with the reduced operating costs, a mathematical model is required to test the potential involved by the number of pumps in the network, once in the field would be unsustainable and impractical to perform these same steps of operation. In this way, the methods used by many researchers include regression models, mass balance models and simplified hydraulic models. At the beginning of the 20th century, computational models integrated into program Epanet are used. These models assume a complexity view, which makes visible in many cases the inefficiency and the deficit of problem-solving capacity in a real context. The mass balance model simply relates the existence of pump grouping as responsible for the variation of the water flow into the reservoir, which presupposes that the water level should be above a certain elevation, given the minimum pressure constraints to feed the network nodes.

The ability to solve numerical methods is based on an iterative process which is responsible to solve the system of linear equations defined by a convergence criterion. To find new hydraulic responses in the system some software's solve the problem through the gradient method where linearizes the energy conservation equations [7].

Some hydraulic simulations methods can be used to establish conditions of hydraulic balance. The Hardy Cross method is applicated on the hydraulic problem by the mesh equations in order to reduce the number of global equations. For that happen it would be necessary the iterative process where each equation is solved one by one, instead determine the system equations simultaneously. This numerical method mentions that all head losses

between two junctions in the network must be the same. Hardy Cross method has some disadvantages like the slow rate and lack of convergence [7].
The method developed by Todini and Pilati in 1987, came introducing the balance between the optimization methods and the Newton-Raphson techniques in order to solve the problem of unknown nodal heads and flows in pipes. This assumption was based on a singular model composed of a linear and non-linear parts of the system. Then, Newton Raphson derivation concentrates this flows and heads to prove the singularity of a resultant solution made by this method, that however is reconducted to an algebraic recursive solution of a linear system [8].

The linear system equations have the same size as the column matrix dimension and are composed of unknown nodal heads, as well is made up of a matrix which is a result of the unknown pipe flows.

The regression models can be more accurate than the method described above since the non-linear equations of which are compounds reflect the behavior of the network under a given set of demands [9]. On the other hand, the simplified hydraulic models can acquire a scheduling composed by an equation representing the numerous components of the network. With the evolution of this type of models have emerged naturally more models with improved technology and were able to provide a very effective response in the case of system modifications [9]. In these hydraulic simulation models, the regression and mass balance methods are included.

The main leap in this technological evolution was mainly noted in the creation of artificial intelligence networks, which can relate to a set of functions with the natural physical parameters of the network, all in real time. These data include the number of pumps, number of valves as well as the time intervals measured during operation [10].

In practice with the use of this method, if a pump fails without containing an active standby system, it will be automatically created a backup process that helps the human operator taking account of pump inactivity and in this case previously informed about the best and most effective extent of support for the continued normal network [10].

2.1.2 Numerical programs – Epanet

During the planning and description of a water supply network, it becomes essential to overcome obsolete methods that by virtue of technological advancement fell into disuse. The numerical simulators can create a distribution network wrapped in a robust software that can agglomerate design and modeling network components. In this context, the hydraulic simulators more recognized are Epanet, Branch, LOOP, Aquis, H2O map, WaterCAD, with the first three being available in the public domain, unlike the others that remaining at commercial regime [11].
All these programs have different advantages, being that the free versions have some inefficiencies in comparison with the others software. However, they all present a strong credibility and confidence within the sector, particularly in the case of Epanet.

Another powerful and versatile software is WaterCAD, whose features include graphical components with great quality such as good flexibility in storage files. This software uses also features linked to water quality as well as the incorporation of elements delineated in the AutoCAD program [11]. Another advantage is the inclusion of GIS (Geographic Information Systems) as a platform that enables the capture, analysis, and dissemination of geographic data [2]. This type of interface really helps to understand the conditions and characteristics of the hydraulic system as it provides information about the length of the pipes, the calculation of demand as well aspects linked to geographical elevation [2].

The Epanet is one of the most used hydraulic simulation software in the world, a fact which is explained by its free access, but also because of its robustness that allows design and simulate any type of network [7].
Originally developed in the year 1993 by EPA (U.S. Environmental Protection Agency), the Epanet nowadays is distributed by two interfaces, one is the simulation component that represents the elements of the system and the second component, which is responsible to reproduce the optimization and calibration features. It also allows the combination of several inclusive losses calculation methods such as Darcy-Weisbach, Chezy-Manning, and Hazen-Williams.

2.2 Calibration methods

Calibration of pipe network systems consists of determining the physical and operational characteristics of the network. The calibration methodology is needed to solve the data from the network in order to give solid and feasible results. These results can be predictable as flows and pressures [3]. The procedure suggested to calibrate a model is represented by various steps: (a) as the identification of the pretended model, (b) the determination and collection of calibration parameters and data, (c) the evaluation of the model results and (d) a sensitive analysis of calibration [5]. Many calibration procedures were developed since the 1970s and since then these can be grouped into three categories, I) the iterative procedure that corresponds to the classical methods, II) the hydraulic simulation models related with explicits methods and III) the optimization models. The first group, developed in the 80s, uses the trial and error of unknown parameters like heads and flows to obtain a steady-state mass balance and energy equations. This procedure was conducted to decrease the model's errors, although the main issue continues to represent a limitation on the system, because of it modeler experience [5].

According to the Fig. 1, the trial and error model explain an iterative way to solve a group of equations, which can be applied for obtaining the least square correction given by different qualities of observations involved. In the case of a determined model, this method is highly effective and offer a computational advantage because there is no requirement to perform the data computation repeatedly. The nonlinear energy equations can be calculated by the Newton iterative method given the corrections of the unknown pipe flow and nodal head allowing the convergence of the model. The pipe flows and heads can be obtained by solving the set of steady-state mass balance and energy equations. The fact of having an iterative procedure causes a higher resolution time than other methods.

The second lately investigated is based on solving the same steady-state equations by knowing the heads and flows measurements in order to calculate the equations constraints.
This method shows a few limitations of equalizing the number of unknown calibration parameters with the available measurements to get a determined solution of equations system without including the error of measurements that are considered 100% accurate. Another disadvantage is at the calibration problem who needs to be decoded even before been determined.

The third group related with the optimization models uses an objective function that minimizes the difference between measured data and the theoretical measurements of the water supply network variables based on the theory of implicit models. Normally is used the least squares method as a criterion of minimization of the objective function, trying to minimize the sum of the squares of the differences between the variables considered [5]. The existence of more than a minimum inherent to the objective function affects the accuracy and precision of the process since the software selects a local minimum of the function, instead of

the global minimum. One of this implicit approach used a more general model for optimal design and operation of water distribution system, which combines the generalized reduced gradient method and penalty methods through the optimization method [5].

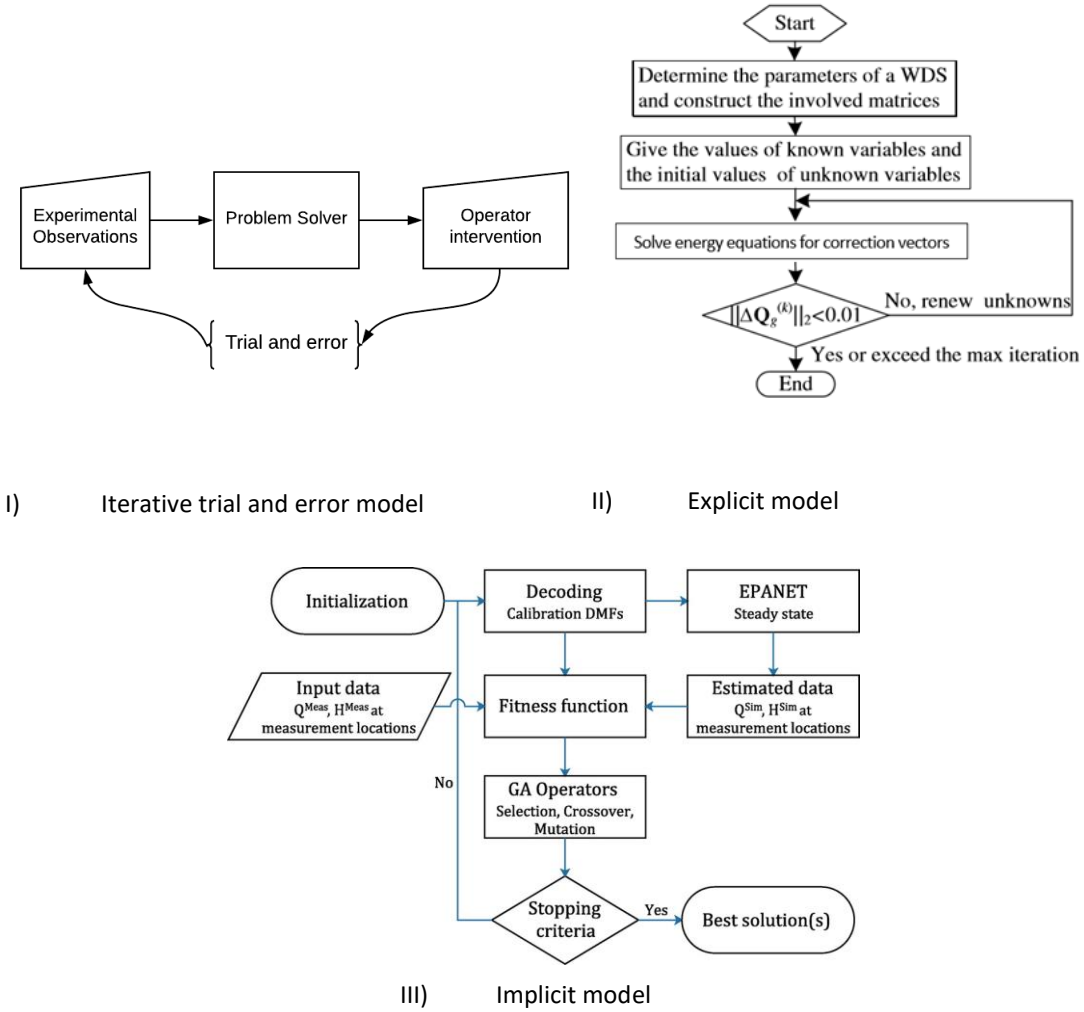


Fig. 1. Scheme of some classical numerical methods. II) retrieved from [18], III) retrieved from [17]

The main difference between explicit and implicit models is the application of the observations. Implicit models can use all the measurements collected, instead of the needed to have many state variables as many measurements made.

2.2.1 Direct problem and Inverse problem

These two principles are an example of complementarity and transversally condition to solve the calibration problem considering the surrounding parameters. The Direct problem is created with the purpose of obtaining some dynamic variables of the network, knowing some initial parameters like roughness, pump curves or other variables. The Inverse problem is

simply the opposite principle, where the knowledge of the field values, normally are heads or flows, or both, allows the method fill in the corresponding parameters that characterize the system. In this water systems, the known parameters including in the Direct method usually are pipe roughness, pump curve, and pipe length. At this point, the heads and flows are determined by the Direct method. The Fig. 2 and Fig. 3 explain through the flowchart the connection between the elements of the calibration process including the Direct and Inverse problem.

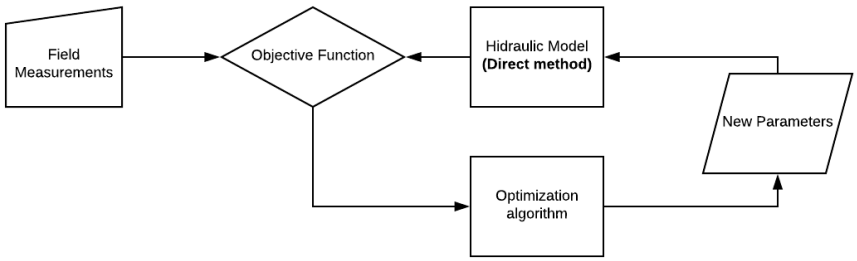


Fig. 2. Scheme of a Classical Inverse method

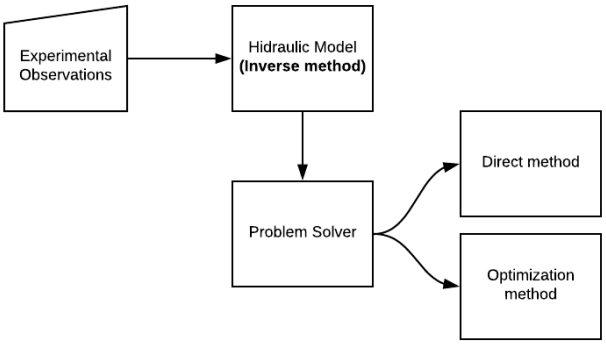


Fig. 3. Scheme of an Inverse explicit method

In the direct problems, the rigorous knowledge of the prior information results in obtaining respective effects estimation. In the inverse problems, the measurement and verification of the effects allow estimating its causes. The Inverse problem is normally defined by knowing as initial parameters the Heads and flows looking to obtain nodal demand or pipe constituents values.

2.3 Classical methods

Errors are, in most of the times the main problem in mathematical models application due to the incorrect values of some collected data but can be minimized through the process of model calibration. One of the solutions considered in the past was to solve the problem of nonlinear optimization. This approach can bring the advantage on loading multiple conditions as the direct incorporation of pressures, flow rates, tank levels into the calibration process [12].

Another implicit model, in 1991, with the conclusion of limited data used during the case studies tests was created. In this case, it was crucial understanding when there are enough parameters to be collect and give them the confidence to solve the equations system. Later, another problem related to the detection of leaks has emerged. In this way was assumed that leaks could have an origin in the network nodes and need to be calculated. For this to happen, it's required that leaks would be part of an equation composed by the equality of the product of the constant discharge coefficient and the leak area, using simply heads and flows as inputs [5].

Other approach surged in the 90's, this time with recourse to the Genetic Algorithm technique. This approach uses the least square type as numerical method resolution that combines three loading conditions like minimum, maximum, and average demand hour. This type of evolutionary method triggered the easy addition of calibration parameters which can be part of the calibration model. Examples of these are the cases of nodal demands and pipe diameters insertion [5].

This kind of method was a precursor in such a way as to find the objective function of the least square type and minimize to a set of implicit and explicit constraints, recreating a new robust and agile optimization technique.

The Todini approach came to introduce a new experimental way of calibrating a multiple loading conditions of the steady-state Water Distribution Systems model, using only pipe roughness coefficient as input. With this statement, it was easier to shape a global understanding of the network because of the linearity of the proposed method [5]. The major advantage of the proposed method is its linearity which is used to solve the problem [5]. This method was helpful for the analysis of a network and avoid the reducing of variables on multiobjective optimization that apply iterative problem resolution. Using this optimization process allowed shrunk the model size without compromising the integrity of water and energy balance requirements in the water distribution systems [13].

The formulation problem of the model calibration could be intended as a first step to clarify errors and discrepancies in the data collected. This differences between measured values and the model predicted values are called "residuals". As seen in Fig. 3, the global movement works by an iterative process, beginning at the field measurements including length, diameter, and roughness of the pipe. Next, they are incorporated directly into the hydraulic model which enable the decision variable of the optimization algorithm in order to discover the network behaviors like heads and flows. This implicit model could calibrate linearly the network in only one iteration, if the known conditions available, are equal or higher than unknown parameters, if not they are solved throw the optimization [14].

2.3.1 Heuristics methods

Hydraulic models can be analyzed according to a sampling design of their calibrations processes. The purpose of monitoring the characteristics of a water distribution system can be described as a research to reach the objectives and a compliance evaluation with the required system performance where can be included leaks.

The heuristics theories are almost used to demonstrate a quick and easy solution created by practical methods with sufficient evidences. Some of the sampling design of model calibration for water distribution systems contain a few problems to get reliable results, so a sensitivity heuristic procedure shows that a minimization of a model prediction, based on a Jacobian matrix can estimate heads and flows by finite difference approximations. There is a weakness associated with this methodology that is the complexity of using big grids to

applicate this kind of methods and testing that type can bring some additional aspects to develop in order further to test methodology [13].

2.3.2 Numerical methods using optimization

According to Ormsbee et al., the incorporation of conservations of mass and energy equations don't use at beginning the optimization process but firstly is provided by an evaluation of the hydraulics parameters [12]. The next step though was to revalidate the optimization routine as the new solution obtained. According to the objective function below, that can update decision variables and evaluates the constraints of the system unleashed by the optimization algorithm. The final procedure takes into account the decision variables that pass again by the simulation part where are repeated until the calibration process obtained be decent.

Using the mathematical notation, the pipe network calibration problem may be summarized as follows:

$$\begin{aligned} &\text{Minimize:} && f(\tilde{\mathbf{X}}) \\ &\text{subject to:} && g(\tilde{\mathbf{X}}) = 0 \\ & && L_h \le h(\tilde{\mathbf{X}}) \le U_h \\ & && L_x \le \tilde{\mathbf{X}} \le U_x \end{aligned} \tag{1}$$

where the optimization algorithm generates a set of calibration parameters like pipe roughness, hydraulic grade line, and flowrate, that are passed to the simulation program. It is described as a function where $\tilde{\mathbf{X}}$ is the decision variable, $f(\tilde{\mathbf{X}})$ is the nonlinear objective function, $g(\tilde{\mathbf{X}})$ is a vector of both linear and nonlinear equality constraints. The vector $h(\tilde{\mathbf{X}})$ represent the nonlinear inequality constraints, supported by L_h, $L_{X,}U_h$, and U_x which are the lower and upper bounds on the decision variables. The several nonlinear optimization problems as formulated by Eq. (1) may be solved using the gradient method or direct search methods [12].
Mathematically, the optimization can be described as the minimization of the objective function $f(\tilde{\mathbf{X}})$ subject to constraints $g(\tilde{\mathbf{X}})$.

The use of the hydraulic model for the objective function can bring in some cases a difficulty to estimate partial differentials equations of the design variables. In order to make the optimization problem simpler is used meta-heuristic optimization methods, with such constraints included in the objective function as new operating conditions. These conditions could be water production costs, pumping energy costs and other design variables [15].

2.3.2.1 Optimization methods

A modified Gauss-Newton called Levenberg-Marquardt method was mentioned to solve several calibrations problems, this gradient type method introduces a new positive constant parameter to all main diagonal terms in the half Hessian matrix.

The advantage related to the quickness of the numerical approach make this traditional method a powerful example on leading the function evaluation in a local search, however bigger the optimization function, more difficult and slow will be the application of this method in case of bounded parameters [5].

According to Lingireddy and Ormsbee, two optimization methods to solve a quadratic programming problem were developed, in this case, both methods required the calculation of nodal pressures made by two lines of thought. The first uses a linear programming method and the second called GINO tries to determine the solution through the gradients [16].

Other class of optimization methods, Genetic algorithm, based on natural evolution show how some species adapt to the surrounding environment. The filtered result of the "best" solution is called by "survivor". This evolutionary computation method considers the inclusion of a few convergence problems, which can traduce in the system some complications in order to find an optimum criterion to define the required population [5].

One of the main advantages of using genetic algorithms to calibrate water distribution systems it relates to the search technique, allowing multiplies directions of the required information. This statement compared with classical methods brings an evolution of model search facilities, in the case of sampling design these improvements could simplify some issues of the model calibration process. Other heuristic method proposed that includes aspects related to this work, is the use of optimization algorithms information for the pump scheduling [9]. The influence of the heuristic information is controlled by a parameter saying the higher is the value, largest is the influence on the algorithm decision. This characteristic represents an estimation of the benefit of choosing one variable over the other alternatives. In this case, this heuristic information based on schedules with low energy cost, have long inactive periods and short operating periods. There are some assumptions that need to be taken, like associate system constraints, such as pressure issues, that can make interferences into the simulation process. The schedule would be considered ineffective without a complete simulation. So, the experimental results can conclude if the algorithms are configured with adequate parameter settings. This heuristic information could help the algorithms to reduce energy cost [9].

2.3.2.2 Other methodologies

Lansey et al (2001) as cited by Kapelan (2002) considered the uncertainties of head and flow measurements are related to the results of the calibration process. This kind of sampling design problems could be solved with a genetic algorithm method associated with a hydraulic numerical program like Epanet, according to Meier and Barkdoll (2000) as cited by Kapelan (2002), the optimal solution was compared with the results of the pipe velocities done by Epanet and validated with the search technique of the genetic algorithm process. [5].

2.4 Troubleshooting models

The models used to represent the diverse criteria of a calibration method could be implemented without the gradient model. In this way, the use of an objective function in the optimization procedure brings the power of decision variables that includes several directions of the optimization parameters [16]. A parametric study can discover the influence of population size associated with the performance of the genetic algorithm and be capable to use on optimization framework analysis. This process gives to the calibration model the perception of a near global optimal solution which can be accompanied by a genetic algorithm of a numerical model. Almost every approach of calibration tries to improve the efficiency of the performance of different methods by determination of the system characteristics. The Direct method captures only pressure and flow measurements at strategical nodes in the system, where was optimize the number of junction nodes to reach reasonable measurements. One related obstacle is the unknown water partitions patterns during data

collection, however, during the long process of calibration the roughness parameter was accepted and worked with more accuracy [17].

Another method that results of the indeterminacy parameters is treated in order to compare pressure and flows measured values with the consumption of the supplied area, in this case, was considered the measurements at late night hours to get a real zero consumption [14].

3. Numerical methodology and formulation

The calibration accuracy depends on the density and location of the measurement points, which can be characterized by a lack of real water consumption. This Direct method uses pressure and flows as the measurements of some selected points of the water distribution system. The direct method is used to develop an alternative framework. In this methodology, the efficiency of the calibration depends on different approaches of the method performance, which are classified as classical methods. Since there are a few issues to measure pipe roughness and nodal demands because they are responsible for many uncertain parameters it is important to choose the best hydraulic method to solve the general problem. In the case of nodal demands and pipe roughness calibration, in which heads and pipe flows variables are assumed as known, it makes sense assume the inversion problem as the way of the model construction.

To better understand the purpose of calibrating a hydraulic network model, it is important to identify the intended use of the model. In general, most of calibrations methods are often used for operational studies, design projects, or just to develop data related to hydraulic performance analysis.

3.1 Direct method for hydraulic systems

To apply this type of method is needed, firstly, define the matrix **A** (i, j) which describe the relationship between fixed head nodes belonging to each pipe. The main problem is to determine all the flow rates (**q**) and all of the unknown heads (**H**) at the nodes following the steady state assumption. Todini and Pilati (1987) considered the importance of required conditions for the steady-state flow to simplify the simultaneous achievement of nodal balance and the head loss-flow relationship expressed in Eq. (3). The model prediction condition can be described as the following system of mass balance and energy equations triggered by $\mathbf{Aq} - \mathbf{M} = 0$ and $\mathbf{A}^{\mathbf{T}}\mathbf{H} + \mathbf{h} + \mathbf{A}_{10}\mathbf{H}_0 = 0$,

$$\begin{bmatrix} \mathbf{A} & 0 \\ \mathbf{R}\,|\mathbf{q}|^{e-1}_{diag} & \mathbf{A}^{T} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{q} \\ \mathbf{H} \end{bmatrix} + \begin{bmatrix} -\mathbf{M} \\ \mathbf{A}_{10}\mathbf{H}_0 \end{bmatrix} = 0 \tag{2}$$

$$\mathbf{h} = \mathbf{R}\,|\mathbf{q}|^{e-1}_{diag}\,\mathbf{q}, \tag{3}$$

and where the matrix A elements are represented by

$$\begin{cases} 1 & \text{if flow of pipe i enters node j} \\ 0 & \text{if pipe i is not connected to node j} \\ -1 & \text{if flow of pipe i leaves node j.} \end{cases}$$

From a point of view of matrix formulation, **A** is the incidence matrix (n x m), $\mathbf{A}^{T}$ is the transposed matrix **A** (m x n) where the elements in $\mathbf{A}_{10}$ are determined by the same method used for **A,** describing the connection between the fixed head node and nodal demand; **q, M** are the vectors of pipe flow and nodal demand, respectively; **H,** $\mathbf{H_0}$ are the vectors heads, respectively; **R** is the pipe friction parameters vector (Hazen Williams Head loss); e is the flow exponent and n,m are the number of nodes and pipes, respectively.

Eq. (2) is viewed as the forward model of the Water Distribution Systems introduced by Todini and Pilati (1987) [8]. The nodal demand **M** and the $\mathbf{H}_0$ are the known parameters of the model

and have the mission of executing pipe flow **q** and nodal **H**. The elements in $\mathbf{A}_{10}$ are selected by the relationship of the fix head node, meaning the elevated tank. The number of iterations, *k*, is applied in each situation of the forward model included in

$$\mathbf{J}_{q\text{-}h}^{(k)} = \begin{bmatrix} \mathbf{A} & 0 \\ e\mathbf{R}|\mathbf{q}|_{\text{diag}}^{e-1} & \mathbf{A}^{\mathsf{T}} \end{bmatrix} \tag{4}$$

where the Jacobian has the size of (n + m) x (n + m). To solve this model by Todini approach the residuals linked to the pipe flow and nodal head can be solved by the Newton iterative method given by

$$\begin{bmatrix} \Delta\mathbf{q}^{(k)} \\ \Delta\mathbf{H}^{(k)} \end{bmatrix} = \left[\mathbf{J}_{q\text{-}h}^{(k)}\right]^{-1} \cdot \begin{bmatrix} \mathbf{M} - \mathbf{A}\mathbf{q}^{(k)} \\ -\mathbf{A}_{10}\mathbf{H}_0 - \mathbf{A}^{\mathsf{T}}\mathbf{H}^{(k)} - \mathbf{q}^{(k)}\mathbf{R}|\mathbf{q}|_{\text{diag}}^{e-1} \end{bmatrix}. \tag{5}$$

Once the energy equations are nonlinear, the iterative method can solve the problem after several iterations. The objective of the problem is next solved by the iterative way until minimize

$$\Delta\mathbf{M}^{(k)} = \mathbf{M} - \mathbf{A}\mathbf{q}^{(k)} \quad \text{and} \quad \Delta\mathbf{h}^{(k)} = -\mathbf{A}_{10}\mathbf{H}_0 - \mathbf{A}^{\mathsf{T}}\mathbf{H}^{(k)} - \mathbf{q}^{(k)}\mathbf{R}|\mathbf{q}|_{\text{diag}}^{e-1}, \tag{6}$$

simultaneously within given nodal demand and pipe roughness working under the laws of mass and energy conservation.

Assuming, $\mathbf{X} = [e\mathbf{R}|\mathbf{q}|_{\text{diag}}^{e-1}]^{-1}$, the residuals for pipe flow and nodal head can be found using the method of block matrix inversion derived from the inverse Jacobian, where

$$\begin{bmatrix} \Delta\mathbf{q}^{(k)} \\ \Delta\mathbf{H}^{(k)} \end{bmatrix} = \begin{bmatrix} \mathbf{X}\mathbf{A}^{\mathsf{T}}(\mathbf{A}\mathbf{X}\mathbf{A}^{\mathsf{T}})^{-1} \\ -(\mathbf{A}\mathbf{X}\mathbf{A}^{\mathsf{T}})^{-1} \end{bmatrix} \Delta\mathbf{M}^{(k)} + \begin{bmatrix} \mathbf{X} - \mathbf{X}\mathbf{A}^{\mathsf{T}}(\mathbf{A}\mathbf{X}\mathbf{A}^{\mathsf{T}})^{-1}\mathbf{A}\mathbf{X} \\ (\mathbf{A}\mathbf{X}\mathbf{A}^{\mathsf{T}})^{-1}\mathbf{A}\mathbf{X} \end{bmatrix} \Delta\mathbf{h}^{(k)} \tag{7}$$

are the corrections for pipe flow and nodal head. The development of the iterative process can follow more specifically as follows:

$$\begin{aligned} &\mathbf{q}^{k+1} \rightarrow \mathbf{q}^{k} + \Delta\mathbf{q}^{k} && \Delta\mathbf{M}^{(k+1)} = \mathbf{M} - \mathbf{A}\mathbf{q}^{(k+1)} \\ &\mathbf{H}^{k+1} \rightarrow \mathbf{H}^{k} + \Delta\mathbf{H}^{k} && \Delta\mathbf{h}^{(k+1)} = -\mathbf{A}_{10}\mathbf{H}_0 - \mathbf{A}^{\mathsf{T}}\mathbf{H}^{(k+1)} - \mathbf{q}^{(k+1)}\mathbf{R}|\mathbf{q}|_{\text{diag}}^{e-1}. \end{aligned} \tag{8}$$

After iterations, the optimization problem will be unlocked due to the convergence of demand residuals **ΔM** and head loss residuals **Δh** and once the minimum solution is verified, the hydraulic balance of the network is completed.

3.2 Classical Inverse method

This process named as Classical Inverse method combines not only the Direct method but also the optimization process. This Classical method has a different approach comparing with the Direct method by taking advantage of the optimization solver. The Direct method helps only explain and formulate the problem in terms of initial conditions and variables used and also to support the law of mass and energy conservation through the process. The calculation of

heads and flow rates to initiate the calculation of nodal demands is the first step of the calibration process.

Once the optimization process is started, as the calculations are performed, the information about each iteration of the process is obtained. After seven iterations this method gave an interesting perspective about a robust behavior when is processing different operational situations. In this case, a basic network was used, containing various possible input values as four nodes and five pipes included in the gradient method algorithm. The flowchart is explained in Fig. 4. and the initial solution is given as a trial solution and enter into the Direct method. In that way, all the unknown's heads and flows are finally known after each iteration in order to execute the objective function.

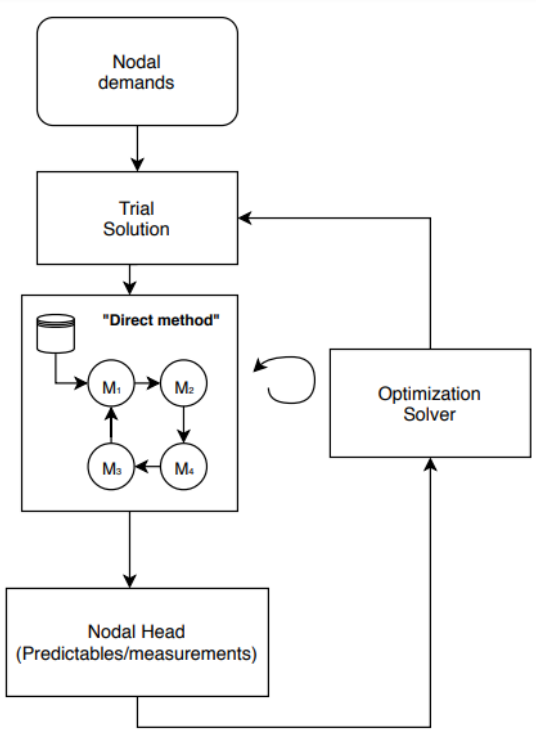


Fig. 4. Classical Inverse method scheme

In this simulation have been included nodal Heads treated as measured and predictable values. In this context can be used as function variables, pipe flows or another known parameter of the network. At the same time that flows and heads are run in each iteration, the nodal demands are calibrated in conformity with the optimization process through an objective function. To minimize the differences between model predictions and measurements this function leads to find out an objective function to establish the error between experimental and predicted values, where is described as

$$\text{min} \quad f(\mathbf{X}) = E^{min} = \Sigma\,(H_i^0 - H_i^p)^2 + \Sigma\,H_e \tag{9}$$

$$\text{Subject to} \quad \mathbf{X}_i^{min} \leq \mathbf{X}_i \leq \mathbf{X}_i^{max}, \qquad i = 1, \ldots, n$$

In this context was assumed Nodal Heads as indicators of predicted values and measured values to accomplish the purpose of the objective function in this method. With this assumption, the validation process is concluded, and all the Heads and pipe flows present as

experimental parameters are considered field measurements with a free error for data comparison [15].

3.3 Inverse model for nodal demand calibration

Thus, following some initial steps to understand which processes matter for the Direct method, there are now conditions to move forward to the new Inverse problem.

The main difference between these two types of methods is the assumption of the measured data, taken as known variables and the unmeasured data as unknown variables. Included in the model construction are the constituents as pipe flows, nodal heads and nodal demands that are implicitly related with the known and unknown variables. The energy equation is presented in Eq. (11), where combined with Eq. (3) is very close to representing the entire Inverse model equations system. The following equations are composed by variables showed in Table 1, where the known and unknown variables should be separated to find the best model representation.

Table 1. Scheme of Inverse model variables

$\mathbf{M}_{kM}$,$\mathbf{M}_{uM}$	Known and unknown nodal demands (subvectors of **M**);
$\mathbf{I}_{kM}$, $\mathbf{I}_{uM}$	Known and unknown nodal demands columns (submatrices of identity matrix $\mathbf{I}_M$);
$\mathbf{q}_{kq}$,$\mathbf{q}_{uq}$	Known and unknown pipe flows (subvectors of **q**);
$\mathbf{A}_{kq}$,$\mathbf{A}_{uq}$	Known and unknown pipe flows columns (submatrices of identity matrix **A**);
$\mathbf{H}_{kH}$, $\mathbf{H}_{uH}$	Known and unknown nodal heads (subvectors of **H**);
$\mathbf{A}^T_{kH}$,$\mathbf{A}^T_{uH}$	Known and unknown nodal heads columns (submatrices of identity matrix $\mathbf{A}^T$);
$\mathbf{h}_{kh}$,$\mathbf{h}_{uh}$	Known and unknown pipe head loss (subvectors of **h**);
$\mathbf{I}_{kh}$, $\mathbf{I}_{uh}$	Known and unknown pipe head loss (submatrices of identity matrix $\mathbf{I}_h$);
kH, uH	Number of Known and unknown nodal heads;
kh, uh	Number of Known and unknown pipe head loss;
kM, uM	Number of Known and unknown nodal demands;
kq, uq	Number of Known and unknown pipe flow;

Eq. (10) describe the continuity equation of the separated known and unknown variables as

$$\mathbf{Aq} - \mathbf{M} = 0 \quad \text{and} \quad [\mathbf{I}_{kM} \quad \mathbf{I}_{uM}] \cdot \begin{bmatrix} \mathbf{M}_{kM} \\ \mathbf{M}_{uM} \end{bmatrix} - [\mathbf{A}_{kq} \quad \mathbf{A}_{uq}] \cdot \begin{bmatrix} \mathbf{q}_{kq} \\ \mathbf{q}_{uq} \end{bmatrix} = 0, \tag{10}$$

$$\mathbf{A}^T_{kH}.\mathbf{H}_{kH} + \mathbf{A}^T_{uH}.\mathbf{H}_{uH} + \mathbf{I}_{uh}.\mathbf{h}_{uh} + \mathbf{I}_{kh}.\mathbf{h}_{kh} + \mathbf{A}_{10}\mathbf{H}_0 = 0 \tag{11}$$

where, furthermore, the Eq. (10) and (11) can be rewritten as

$$\begin{bmatrix} \mathbf{I}_{uM} & -\mathbf{A}_{uq} & 0 \\ 0 & \mathbf{I}_{uq}\mathbf{R}|\mathbf{q}_{uq}|^{e-1}_{diag} & \mathbf{A}_{uH}^{T} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{M}_{uM} \\ \mathbf{q}_{uq} \\ \mathbf{H}_{uH} \end{bmatrix} + \begin{bmatrix} \mathbf{I}_{kM}\mathbf{M}_{kM} - \mathbf{A}_{kq}\mathbf{q}_{kq} \\ \mathbf{A}_{kH}^{T}\mathbf{H}_{kH} + \mathbf{I}_{kh}\mathbf{h}_{kh} + \mathbf{A}_{10}\mathbf{H}_{0} \end{bmatrix} = 0. \quad (12)$$

The inverse problem represented in Eq. (12) needs to follow an approach based on few essential rules to run this model, which one of them is the number of number of measurements that is greater than the unknown nodal demands. With this assumption, the model can obtain directly the unknown nodal demand, while not forgetting the undetermined dimension of the model, with the size of (uM + uq + uH) − (n + m) = uM − kq − kH.

This model construction based on Kun.D et al (2015) refers as model solution the approach of reduce the calibration process dimension by grouping nodal demands. If the model contains many nodes and nodal demands, this aggregation process could be very useful and can be grouped in classes with the same demand pattern [18]. In this case, there is the advantage of eliminating multiple parameters as the screening of measurements and making the calibration easier and fast, but on another hand the parameter uncertainty will increase.

The main goal of this process allows the reduction of the number of unknowns inside the model. The previous framework is accompanied by a demand allocation matrix, where $\mathbf{G}_d$ represent the allocation content of nodal demand:

$$\mathbf{I}_{uM}\mathbf{M}_{uM} = \mathbf{G}_d\mathbf{M}_g. \quad (13)$$

To collocate the model determined, the number of measurements need to be equal or greater than the number of the grouped demand nodes as indicated in Eq. (14), which represent the combination of Eq. (12) and Eq. (13). $\mathbf{M}_g$ can be calculated by summing the individual nodal demands in each group, where $\mathbf{G}_d$ elements can be found by a ratio between nodes base demand and the groups final base demand; g = number of groups. One problem created in this aggregation is the fact of $\mathbf{G}_d$ elements are constant, and the nodes grouped share the same demand pattern in the whole process. For Eq. (13) make sense the expression (n + m) − (g + uq + uH) = kq + kH - g ≥ 0 where it needs to be verified to ensure the balance of the equation. The model is then determined or overdetermined and solvable.
Based on Eq. (13), the Eq. (12) can be updated as

$$\begin{bmatrix} \mathbf{G}_d & -\mathbf{A}_{uq} & 0 \\ 0 & \mathbf{I}_{uq}\mathbf{R}|\mathbf{q}_{uq}|^{e-1}_{diag} & \mathbf{A}_{uH}^{T} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{M}_{g} \\ \mathbf{q}_{uq} \\ \mathbf{H}_{uH} \end{bmatrix} + \begin{bmatrix} \mathbf{I}_{kM}\mathbf{M}_{kM} - \mathbf{A}_{kq}\mathbf{q}_{kq} \\ \mathbf{A}_{kH}^{T}\mathbf{H}_{kH} + \mathbf{I}_{kh}\mathbf{h}_{kh} + \mathbf{A}_{10}\mathbf{H}_{0} \end{bmatrix} = 0. \quad (14)$$

The demand allocation matrix for the Case study network is defined as

$$\mathbf{G}_{d\,=} = \begin{bmatrix} 0.7 & 0 \\ 0.3 & 0 \\ 0 & 0.7 \\ 0 & 0.3 \end{bmatrix}. \quad (15)$$

One important step after Eq. (14) is the determination of the Jacobian matrix where includes unknown nodal demand, pipe flow, and nodal head. The Gauss-Newton iteration method is used to calculate the corrections parameters according to

$$\mathbf{J}_{M\text{-}q\text{-}h}^{(k)}=\begin{bmatrix}\mathbf{G}_d & -\mathbf{A}_{uq} & 0\\ 0 & \mathbf{I}_{uq}e\mathbf{R}\left|\mathbf{q}_{uq}^{(k)}\right|_{diag}^{e-1} & \mathbf{A}_{uH}^{T}\end{bmatrix} \tag{16}$$

and,

$$\begin{bmatrix}\Delta\mathbf{M}_{g}^{(k)}\\ \Delta\mathbf{q}_{uq}^{(k)}\\ \Delta\mathbf{H}_{uH}^{(k)}\end{bmatrix}=\left[\mathbf{J}_{M\text{-}q\text{-}H}^{(k)\ T}\quad\mathbf{J}_{M\text{-}q\text{-}H}^{(k)}\right]^{-1}\left[\mathbf{J}_{M\text{-}q\text{-}H}^{(k)}\right]^{T}\begin{bmatrix}\Delta\mathbf{M}^{(k)}\\ \Delta\mathbf{h}^{(k)}\end{bmatrix}. \tag{17}$$

The Jacobian matrix uses the decomposition from Eq. (14), where residuals from nodal demand, pipe flow and nodal head are multiplied by the system of equations created in each iteration and which the sum of the parts is approximately zero. According to Eq. (17), the unknown nodal demand, pipe flow and nodal head are iteratively updated after iterations, where,

$$\Delta\mathbf{M}^{(k)}=-\mathbf{I}_{kM}\mathbf{M}_{kM}-\mathbf{G}_d\mathbf{M}_g{}^{(k)}+\mathbf{A}_{kq}\mathbf{q}_{kq}+\mathbf{A}_{uq}\mathbf{q}_{uq}{}^{(k)}\ \text{and}$$

$$\Delta\mathbf{h}^{(k)}=-\mathbf{A}_{kH}^{T}\mathbf{H}_{kH}-\mathbf{A}_{uH}^{T}\mathbf{H}_{uH}{}^{(k)}-\mathbf{I}_{kh}\mathbf{h}_{kh}-\mathbf{A}_{10}\mathbf{H}_0-\mathbf{I}_{uq}\mathbf{R}\left|\mathbf{q}_{uq}^{(k)}\right|_{diag}^{e-1}\mathbf{q}_{uq}^{(k)} \tag{18}$$

represent the demand residuals in each node and the head loss residuals in each pipe, respectively.

The final model simplification of Inversion model is also very similar with the Direct method because it can be used the block matrix inversion method. In terms of engineering, its crucial to understand why these methods are used in calibration of Water Distribution Systems and even more important it is their validation. The friction head loss through a pipe can be calculated by the well-known Hazen–Williams formula, where

$$h_f=\frac{10.44Lq^{1.85}}{C^{1.85}\cdot d^{4.8655}} \tag{19}$$

and C is the Hazen–Williams coefficient, L is the length of pipe (ft), the diameter of the pipe, d (in), and q is the flow rate of water (gallons/min).

4. Case study

In the case study, it is considered a basic hydraulic system composed of 4 nodes, 5 pipes, and 1 tank. The values of the Hazen-Williams coefficient (C_{H-W}) are 90 according to the pipe material and age, with the length of 1640 ft and pipe diameter of 7,874 inches. The hydraulic network is presented in Fig. 5. The elevated water tank is connected to pipe 1. All the parameters like pipe roughness coefficient, lengths and diameters are assumed to be known with small errors. Other issues concerning optimal meter placement and sampling design are not within the scope of this study.

The purpose of the Case study is to find the most appropriated values of nodal demands of the network under several conditions. These conditions are related with two approaches: (i) Steady-state and (ii) Transient operations.

The conditions at the Steady-state regime defines the water height of the tank as constant. The time-lapse for the Steady-state Case is not relevant because the operational conditions are maintained as constant over the time.

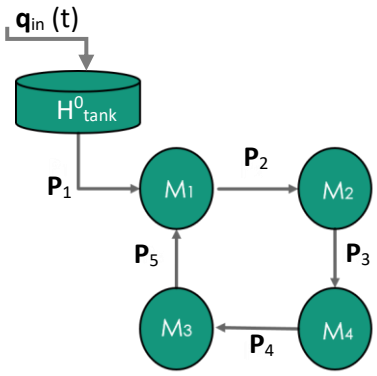


Fig. 5. Case study network

The Transient case simulates another type of conditions that lead to a different scope and different results. To better understand how the variables of the network are integrated on the calibration methodology and which of them are taken as input or output, Table 2 is created. Consider that the water level of the tank for a period of 24 hours can change over time, (H^{i+1}_{tank}) and is identified as $H^{i+1}_{tank} = H^{i} + q^{i}_{in}\Delta t/A - q^{i}_{1}\Delta t/A$, where the flow for pipe 1 (q_1) is changing and is updated at each time step. That change in flow can be observed in the Fig. 6, in which the values are predefined for the calibration method. It is considered that H^{0}_{tank} = 2529 ft, representing the initial height of water tank over the 24 hours. The area is A = 5808,8 ft^2 and time lapse interval is Δt = 60 min. The inflow of the tank, (q^{i}_{in}) is assumed as variable over the 24 hours as shown in Fig. 8. The input flow rate for pipe 1, (q_1) is used by Epanet for the calibration methodology as seen in Fig. 6.

The known parameters include the cross sectional area of the tank and the time interval between contiguous measurements. The calibration process tries to find all the outputs of the system described as the unknown's heads and flows as well as the elevated water tank (H^{i+1}_{tank}) at each hour. During the time, the tank water level is depending on the inputs shown in Table 2.

Table 2. Exemplification of the input and output variables of the calibration method

In		Out
C,L,d →	Calibration methodology	→ $H_{2,3,4}(t)$, $q_{2,3,4,5}(t)$
$q^{i}_{in}(t)$, $q_1(t)$ →		→ $\mathbf{M}_{1,2,3,4}(t)$
H^{0}_{tank} ,$H_1(t)$ →		→ $H^{i+1}_{tank}(t)$

For the transient Case study, the Epanet program was used to create the hydraulic scenario and for prior knowledge of the results. Using a known pattern consumption (called the true nodal demands), the Epanet hydraulic simulator creates the hourly flows and heads of the network presented in Fig. 5.

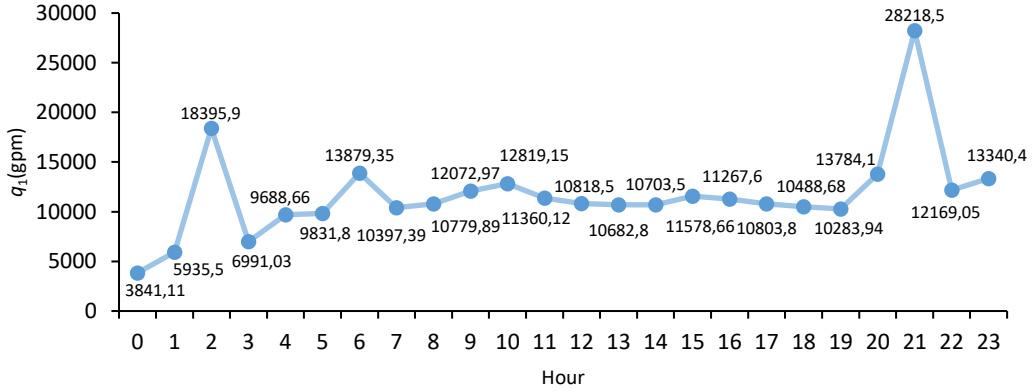


Fig. 6. Flow rate (q_1) of pipe 1 for the Transient case (retrieved from Epanet)

These flows and heads are used as inputs of the calibration methodologies and it is expected that the later methodologies can predict the nodal demands prior known as true nodal demands. In that way, the Epanet is capable to give to the network the needed data for solving the calibration problem as can be seen in Table 3 and 4.

Fig. 7 presents the pattern associated with the present network, in which the hours of more consumption are dimensioned as maximum values. The scale used to represent the height of each column is standardized and has the minimum of 0 and the maximum of 1.

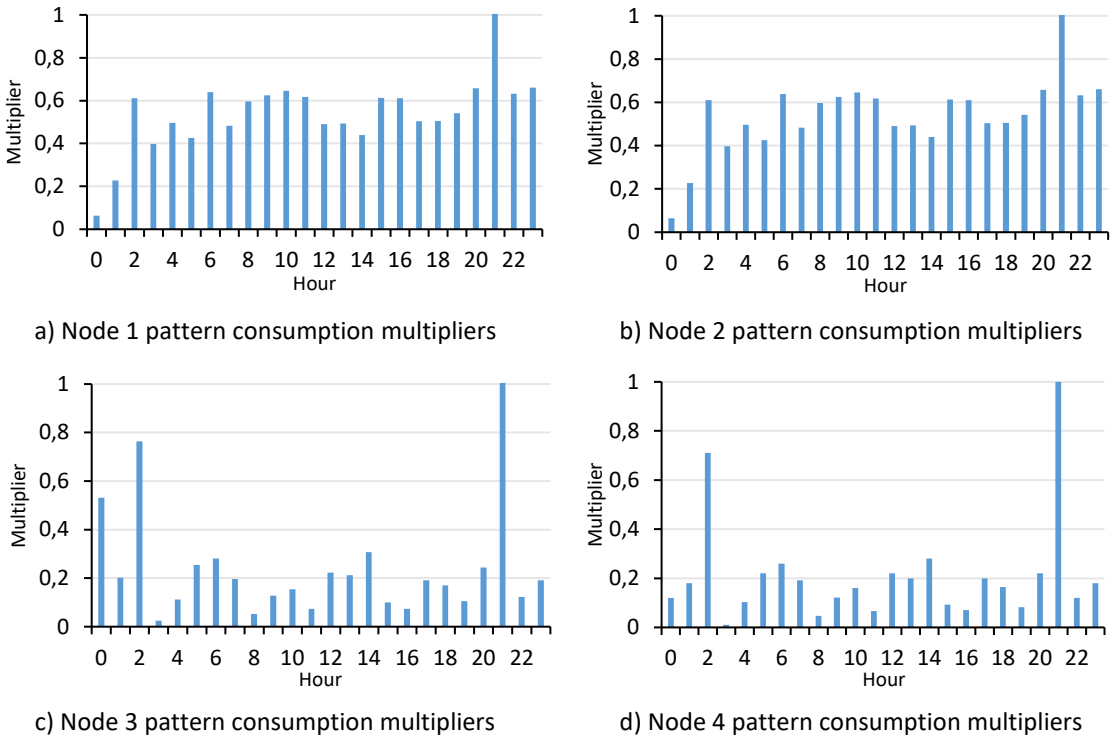


Fig. 7. Pattern consumption multipliers for the Transient Case (retrieved from Epanet)

The maximum consumption at node 1 and node 2 are 12638.0 gpm and 4686.0 at 21:00 h, respectively. For node 3 the maximum value is 3524.5 gpm at 21:00 h and for node 4 is 7370.0 gpm at 21:00 h. For this example, it is used a pattern time step of 24 hours. The inputs by default for Epanet hydraulic simulator are the tank diameter d_t, tank height H^i, pipe length (L), pipe diameter (d) and pipe roughness, (C). The multipliers presented in Fig. 7, m_i (t) multiply by the nodal demand, $M_{i,}$ to know the demand pattern for the transient Case. This pattern associated is assumed as input. The outputs from the Epanet solver are presented in Table 3 and 4 where it is shown the pipe flows and nodal heads for each hour.

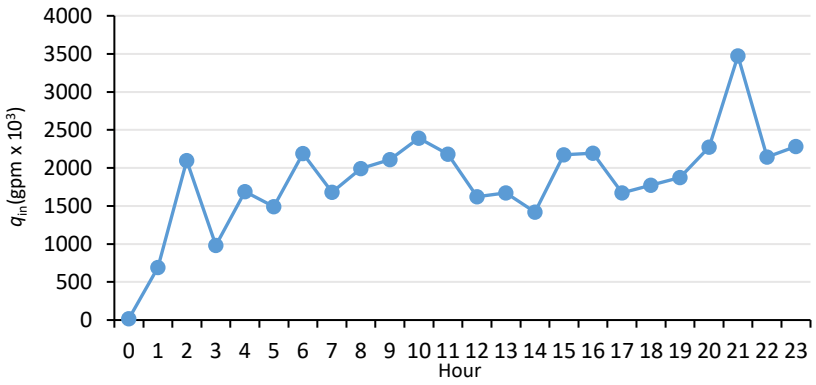


Fig. 8. Flow rate entering into the tank for the Transient case

Table 3. Epanet Heads for the Transient Case (ft)

Hours	Node 0	Node 1	Node 2	Node 3	Node 4
0	2529.00	1753.95	1646.96	1578.64	1580.26
1	8992.00	7256.77	7103.47	7123.22	7082.62
2	24166.00	10067.36	8610.74	8639.89	8244.57
3	15736.00	13386.34	13290.58	13347.3	13315.94
4	19670.00	15369.94	15147.19	15234.18	15156.41
5	16860.00	12441.55	12115.55	12191.49	12102.33
6	25290.00	16922.78	16367.42	16530.07	16362.91
7	19108.00	14207.3	13886.24	13984.65	13884.5
8	23604.00	18364.19	18135.68	18257.35	18176.69
9	24728.00	18265.01	17936.5	18069.76	17953.46
10	25571.00	18348.81	17956.17	18104.46	17964.98
11	24447.00	18672.92	18411.36	18540.32	18447.56
12	19389.00	14114.38	13750.56	13855.64	13744.64
13	19501.40	14348.66	14007.39	14108.67	14004.81
14	17422.00	12250.75	11834.9	11916.01	11801.92
15	24278.40	18296.92	18009.27	18136.68	18034.32
16	24166.00	18478.71	18218.62	18345.62	18252.43
17	19951.00	14689.65	14346.57	14455.83	14345.09
18	20007.20	15026.52	14727.02	14825.57	14727.28
19	21440.30	16638.18	16406.15	16503.93	16424.92
20	26048.70	17787.51	17282.45	17441.35	17282.47
21	39902.00	8763.14	5753.38	6027.45	5166.67
22	25009.00	18450.43	18119.08	18256.01	18137.46
23	26133.00	18357.53	17919.2	18072.11	17923.95

Table 4. Epanet Flows for the Transient Case (gpm)

Hours	Pipe 1	Pipe 2	Pipe 3	Pipe 4	Pipe 5
0	3841.11	1318.61	1021.61	137.21	-1721.5
1	5935.50	1601.21	545.21	-781.39	-1486.29
2	18395.9	5400.26	2562.26	-2670.44	-5341.64
3	6991.03	1241.95	-606.05	-679.75	-765.08
4	9688.66	1959.14	-350.86	-1109.97	-1499.52
5	9831.80	2406.42	426.42	-1194.98	-2085.38
6	13879.35	3208.44	238.44	-1677.76	-2660.91
7	10397.39	2386.69	142.69	-1272.35	-1958.7
8	10779.89	1986.34	-785.66	-1132.05	-1317.55
9	12072.97	2416.40	-487.60	-1379.37	-1824.57
10	12819.15	2660.64	-342.36	-1521.56	-2059.51
11	11360.12	2136.60	-734.40	-1220.82	-1480.52
12	10818.50	2553.35	276.35	-1345.05	-2124.15
13	10682.80	2466.62	176.42	-1297.58	-2039.58
14	10703.50	2744.46	698.46	-1365.14	-2441.04
15	11578.66	2249.16	-602.04	-1287.45	-1639.90
16	11267.60	2130.10	-707.90	-1223.8	-1483.49
17	10803.80	2473.69	130.69	-1343.31	-2011.11
18	10488.68	2298.73	-50.87	-1259.55	-1853.15
19	10283.94	2002.77	-515.13	-1119.47	-1490.47
20	13784.10	3048.11	-10.99	-1632.39	-2485.69
21	28218.50	7991.01	3305.01	-4064.99	-7589.49
22	12169.05	2427.68	-509.32	-1393.72	-1820.38
23	13340.40	2823.63	-245.37	-1571.97	-2239.77

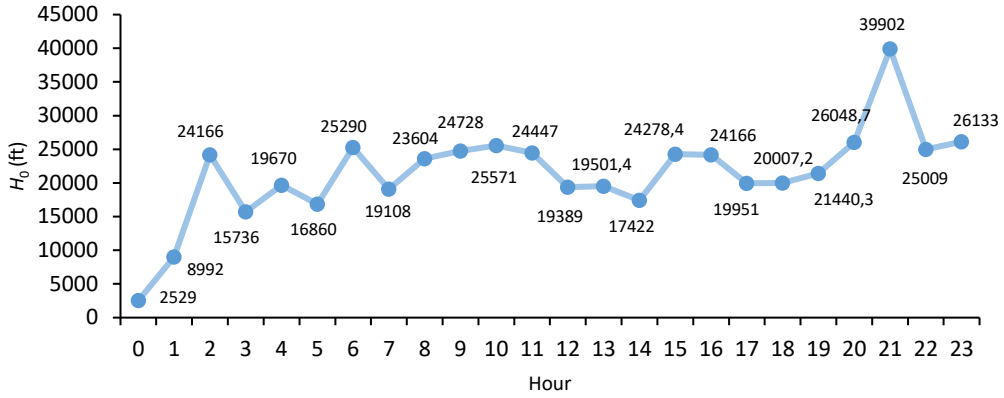


Fig. 9. Nodal heads of node 0 for the Transient case

The initial inflow $q^{i}_{1}(t)$ is assumed as the value generated from the nodal demand calibration process thereby the nodal heads and pipe flows are implicitly known. The pipe balance error is essentially a mass balance through a pipe. Thus, -dE can be defined by the following equation:

$$-dE = (q_1 + q_5) - (q_2 + M_1) \tag{20}$$

where, the pipe flows 1, 2, 5 and demand at node 1, characterize the mass balance in the presented network.
For instance, the mass balance error at 00:00 h based on the initial flow estimates is: (3841.11 + (-1721.50)) – (1318.61 + 801.00) = 0 gpm.

4.1 Numerical procedure

As the intention of the nodal demand calibration is to have a reliable model close to the real values of the network, it is associated with the responsibility of the proposed methods. The Classical Inverse method and the Inversion model for nodal demand are both able to solve a generic problem.
For this Case Study, the two numerical methods are explained below, where is important to present step by step the calibration methodology of the present work.

4.2 Classical inverse method for the steady-state case

In the calibration framework process, both methods can solve the same general problem but with different ways to get all unknown variables. The approach designed by Todini is explored in this first part. Todini uses matrices that contain a set of pipes and nodes equations which represent the directions of the flows between pipes and nodes. With this assumption, the energy balance equations characterize the initial and unknown's variables of the model, in order to make it overdetermined and calculate all the unknown nodal demands. This initial numerical procedure serves as a guide to show the development of numerical work included in the stade-steady and transient regime.

The problem begins to be solvable with the identification of **A** matrix exposed as

Table 5. Incidence matrix A of Classical inverse method

	Pipe 1	Pipe 2	Pipe 3	Pipe 4	Pipe 5
Node 2	0	1	-1	0	0
Node 3	0	0	0	1	-1
Node 4	0	0	1	-1	0

where it represents the connection matrix between pipes and nodes. The matrix **A** and matrix **X** are part of the method of block matrix inversion, where some predefined letters help to simplify the matrix process until getting the Jacobian. In Table 5, the first column shows just the nodes 2, 3 and 4 because node 1 is considered a known variable. This known variable is a Head and has the value 1753.95 ft for the steady-state case. The matrix **X** is shown as

Table 6. Matrix **X** *of* Classical Inverse method

Index	1	2	3	4	5
1	2.69	0	0	0	0
2	0	8.16	0	0	0
3	0	0	7.13	0	0
4	0	0	0	3.29	0
5	0	0	0	0	23.17

where $\mathbf{X} = [e\mathbf{R}|\mathbf{q}|^{e-1}_{\text{diag}}]^{-1}$ to compute the method of block matrix inversion given by

$$\left[\mathbf{J}^{(k)}_{q\text{-}h}\right]^{-1} = \begin{bmatrix} \mathbf{X}\mathbf{A}^{\mathsf{T}}(\mathbf{A}\mathbf{X}\mathbf{A}^{\mathsf{T}})^{-1} & \mathbf{X}\text{-}\mathbf{X}\mathbf{A}^{\mathsf{T}}(\mathbf{A}\mathbf{X}\mathbf{A}^{\mathsf{T}})^{-1}\mathbf{A}\mathbf{X} \\ -(\mathbf{A}\mathbf{X}\mathbf{A}^{\mathsf{T}})^{-1} & (\mathbf{A}\mathbf{X}\mathbf{A}^{\mathsf{T}})^{-1}\mathbf{A}\mathbf{X} \end{bmatrix}. \tag{21}$$

And where the matrix included in the inverse Jacobian is expressed in Table 7.
The global inverse Jacobian presented in Table 9 is a crucial matrix to develop the final procedure for the corrections of pipe flow and nodal head.

Table 7. Matrix $(\mathbf{A}\mathbf{X}\mathbf{A}^{\mathsf{T}})^{-1}$

Index	1	2	3
1	0.097	0.009	0.069
2	0.008	0.040	0.020
3	0.069	0.019	0.150

These corrections are filed at the beginning of each iteration. This process can be executed by the inversion function of Excel. The Jacobian is presented in Table 8, which incorporate the most advanced simplification of the matrix computation, thus allowing the validation of the Eq. (21).

Table 8. Global Jacobian Matrix [$\mathbf{J}^{(1)}_{q\text{-}H}$] of Classical inverse method (first iteration)

Elements	P1	P2	P3	P4	P5	N2	N3	N4
N2	0	1	-1	0	0	0	0	0
N3	0	0	0	1	-1	0	0	0
N4	0	0	1	-1	0	0	0	0
P1	0.371	0	0	0	0	0	0	0
P2	0	0.122	0	0	0	1	0	0
P3	0	0	0.140	0	0	-1	0	1
P4	0	0	0	0.303	0	0	1	-1
P5	0	0	0	0	0.043	0	-1	0

Table 9. Inverse Jacobian $[\mathbf{J}_{q\text{-}h}]^{-1}$ of Classical Inverse method

Elements	N2	N3	N4	P1	P2	P3	P4	P5
P1	0	0	0	2.655	0	0	0	0
P2	0.799	0.071	0.569	0	1.642	1.642	1.642	1.642
P3	-0.201	0.071	0.569	0	1.642	1.642	1.642	1.642
P4	-0.201	0.071	-0.431	0	1.642	1.642	1.642	1.642
P5	-0.201	-0.929	-0.431	0	1.642	1.642	1.642	1.642
N2	-0.098	-0.009	-0.070	0	0.799	-0.201	-0.201	-0.201
N3	-0.009	-0.040	-0.019	0	0.071	0.071	0.071	-0.929
N4	-0.070	-0.019	-0.149	0	0.569	0.569	-0.431	-0.431

Table 10 shows the first iteration of the Classical Inverse method by Todini approach. Others solution for the next iterations can be found in the Appendix A. The solution for the first iteration method uses the initial guesses for pipe flow and nodal head with the respective direction of the flow balance of the network without any requirement.

Table 10. Classical Inverse method first iteration

index	**H| q**	1	2	3	4	5	2	3	4	RHS
2	2790.00	0	1	-1	0	0	0	0	0	996.27
3	1800.00	0	0	0	1	-1	0	0	0	-2366.63
4	1400.00	0	0	1	-1	0	0	0	0	3648.39
1	3821.13	0.377	0	0	0	0	0	0	0	2042.78
2	1036.12	0	0.122	0	0	0	1	0	0	-68.61
3	1214.07	0	0	0.140	0	0	-1	0	1	-91.98
4	3008.00	0	0	0	0.303	0	0	1	-1	-492.81
5	303.61	0	0	0	0	0.043	0	-1	0	-7.08

Each cell of the Jacobian included in Table 10 as top left matrix, accounts for each node in the row, the second cell in the first row accounts for the connectivity of node 2 with pipe 1, the third cell in the first row accounts for the connectivity of node 2 with pipe 3, and so on. The implicit network parameters as diameter, C-factor, and L, are included on the pipe head loss vector expressed in Eq. (4).

The derivatives with respect to the nodal heads are 0, 1, or -1. When 0 is used for a pipe, it means that the pipe is not connected to a node 1 is for a pipe connected to a node and it's called an inflow (sink node), and a -1 is used when a pipe is connected to a node, but the flow is exiting the node (source node) [19].

With the Jacobian matrix and balance error matrix determined, the change in demand/head (RHS) loss matrix can be discovered. This demand/head loss residuals matrix can be determined by multiplying the inverse of the Jacobian matrix by the balance error matrix thus getting

$$\begin{bmatrix}\mathbf{\Delta q}^{(1)}\\ \mathbf{\Delta H}^{(1)}\end{bmatrix}=\begin{bmatrix}5361.04\\ 1618.00\\ 621.73\\ -3026.66\\ -660.02\\ -266.81\\ -21.40\\ -445.94\end{bmatrix} \tag{22}$$

where the corrections for pipe flow and nodal head are based on Eq. (5). The initial pipe flows and nodal heads are represented in Table 11 and a convergence result can be obtained after seven iterations. Once the change in demand/head loss matrix is completed, the initial nodal heads and flows are updated, and the next iteration is performed. This procedure becomes an iterative process with the goal of update flows and heads until a certain minimum of the residuals presented in the balance error matrix are converged.

Table 11. Initial nodal heads and flows (ft | gpm)

index	**H**| **q**
H_2	2790.00
H_3	1800.00
H_4	1400.00
q_1	3821.13
q_2	1036.12
q_3	1214.07
q_4	3008.00
q_5	303.61

The algorithm that includes the Direct method (Classical Inverse method), solved initially on Excel, pointed out the easy way to find out all the unknown flows and heads simultaneously. In this context, the first step was to transform all the unknows flows and heads developing the continuity equations. The residuals from Eq. (5) are converged to zero making the flow and heads found Direct method, as shown in Fig. 10.

At Table 12 it is possible to see the evolution of the hydraulic process on pipe flow 2 and 5, and nodal head 4, one by one iteration.

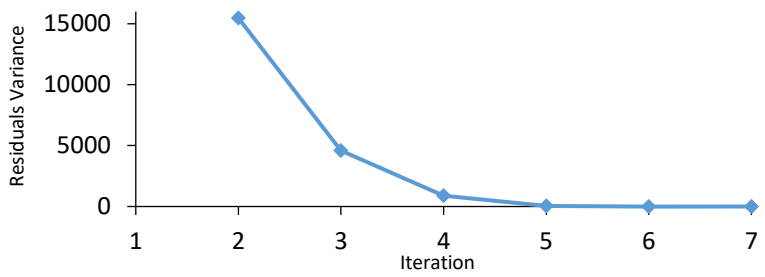


Fig. 10. Dimensionless Residuals convergence of the Direct method in the Classic Inverse Method

Table 12. Iteration sequence for the Todini and Pilati Classical Inverse method formulation

Iteration	q_2 (gpm)	q_5 (gpm)	H_4 (ft)
1	1036.110	303.612	1400.000
2	2654.117	-356.411	954.059
3	1927.897	-1082.632	905.650
4	1600.167	-1410.362	721.970
5	1600.984	-1409.545	333.409
6	1600.984	-1409.545	333.409
7	1600.984	-1409.545	139.125

In Table 13, and after the optimization solver procedure used from Excel, the final error is closer to zero, representing the feasibility between predictable and measured parameters. After this, is possible to understand how the optimization recourse works with the Direct method. The optimization process is originated from the solution of the iterative procedure, which E_{min} is constantly updated from the nodal demands result. In this way the objective function relates directly to improve the performance between the nodal demands result and the heads. In this case, the experimental heads were used to accomplish the heads of the iteration process. For this case could be used flows instead of heads. The Fig. 11 shows the error variation across the iterative process where it is made seven iterations to get the results of the optimized E_{min}. E_{min} is the final sum of the predictable heads. The nodal demands results could be seen in Table 14 and they are generated following all the balance energy equations derived from the hydraulic model.

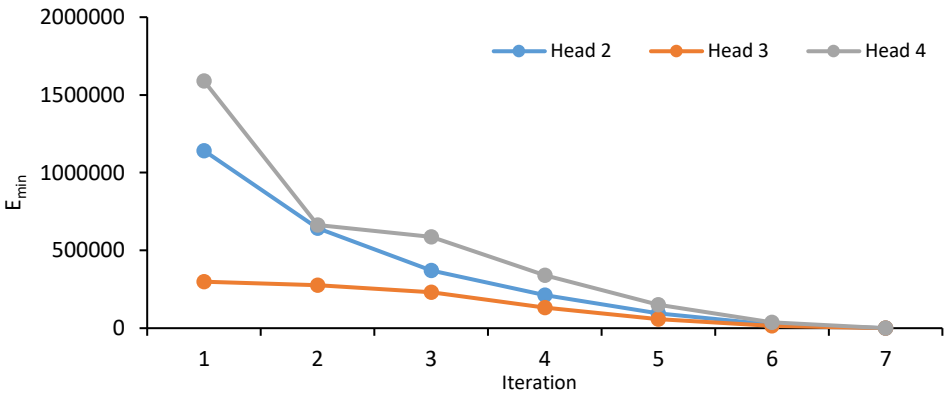


Fig. 11. Convergence of the objective function error from the Direct method

Table 13. Optimization solution of the Classic inverse method

Experimental		Predictable	Residuals	Solver
H_2 (ft)	1721.4300	1721.4288	ΔH_2	5.765 E-06
H_3 (ft)	1253.3400	1253.3410	ΔH_3	1.433 E-07
H_4 (ft)	139.1200	139.1251	ΔH_4	3.918 E-06
E_{min}				9.826 E-06

Table 14. Nodal demands result of the Classic inverse method

nodes	$\mathbf{M}_{uM}$ (gpm)
1	818.31
2	337.75
3	1854.46
4	810.60

These parameters are the final calibrated nodal demands inspired in Todini approach [8], respecting the network from Fig. 5.

4.2.1 Inverse model for nodal demand calibration

A similar method was implemented with the same goal of calibrating nodal demands but without an implicit optimization process. The main differences at the beginning are in terms of indexing vectors and subvectors, which can be classified as known and unknown variables. These variables should be decomposed and respect the continuity equations. In the final procedure of the model, the unknown nodal demands can be obtained directly. In this case, two demands from node 1 and 2, and, node 3 and 4 were grouped in two groups, such that the number of unknowns can be reduced to make the calibration problem solvable. This step helps the model not only solving problems in a few iterations, but also gives more reliable results because some measurements are not available. This aggregating system is a good alternative to achieve an efficient and quick system resolution. The matrix $\mathbf{G}_d$ based on Eq. (14) is a demand allocation matrix with the size of *n* (nodes) x *g* (groups) that is involved in every iteration presented on the Jacobian matrix as seen in Table 15. Table 16 shows the initial unknown and known pipe flows and nodal heads.

Table 15. Global Jacobian Matrix of Inverse model (first iteration)

Elements	G1	G2	P1	P2	P3	P4	P5	N2	N3	N4
N1	0.7	0	1	1	0	0	-1	0	0	0
N2	0.3	0	0	-1	1	0	0	0	0	0
N3	0	0.7	0	0	0	-1	1	0	0	0
N4	0	0.3	0	0	-1	1	0	0	0	0
P1	0	0	0.376	0	0	0	0	0	0	0
P2	0	0	0	0.150	0	0	0	-1	0	0
P3	0	0	0	0	0.121	0	0	1	0	-1
P4	0	0	0	0	0	0.022	0	0	-1	1
P5	0	0	0	0	0	0	0.189	0	1	0

Table 16. Initial unknown and known pipe flows and nodal heads

Pipe	**q**(u*q*) (gpm)	**q**(k*q*) (gpm)	Node	**H**(u*H*) (ft)	**H**(k*H*) (ft)
1	-	3841.11	2	1646.96	-
2	1318.61	-	3	1578.64	-
3	1021.61	-	4	1580.26	-
4	137.21	-			
5	-1721.50	-			

The resolution has a different method when compared with the Todini approach. In fact, that is possible to see in Eq. (17). Instead of having just the multiplication of the Inverse Jacobian by the hydraulic equations as shown in the direct method, the matrixial multiplication to achieving the inverse model residuals is more complex when compared to the Direct method. Tables 17 and 18 help to better understand the procedure of the model resolution.

Table 17. Inverse model Transposed Jacobian solution for the first iteration

index	Transposed Jacobian									
1	0,7	0,3	0	0	0	0	0	0	0	0
2	0	0	0,7	0,3	0	0	0	0	0	0
1	1	0	0	0	0,3761	0	0	0	0	0
2	1	-1	0	0	0	0,15036	0	0	0	0
3	0	1	0	-1	0	0	0,12104	0	0	0
4	0	0	-1	1	0	0	0	0,02197	0	0
5	-1	0	1	0	0	0	0	0	0,18861	0
2	0	0	0	0	0	-1	1	0	0	0
3	0	0	0	0	0	0	0	-1	1	0
4	0	0	0	0	0	0	-1	1	0	1

Table 18. Solution of the (Eq. 17) in the first iteration of Inverse model

index	$[\mathbf{J}_{M-q-H}^{(k)T}\mathbf{J}_{M-q-H}^{(k)}]^{-1}[\mathbf{J}_{M-q-H}^{(k)}]^{T}$									$\Delta\mathbf{M}^{(k)}\Delta\mathbf{h}^{(k)}$		$\Delta\mathbf{M}_{g}^{(k)}\ \Delta\mathbf{q}_{uq}^{(k)}\ \Delta\mathbf{H}_{uH}^{(k)}$	
1	1,21847	0,49023	0,04114	-0,096	-3,2658	-4,84329	-4,84329	6,242274	6,242274	-11,08556		-820,879	8272,074	Δ M1
2	-0,2185	0,50977	0,95886	1,096	0,58556	4,843288	4,843288	-6,24227	-6,24227	11,085562		-377,919	-1343,57	Δ M2
1	7,4E-14	2,7E-14	5,9E-14	5E-14	2,68026	-1,1E-16	1,03E-14	-8,7E-15	-6,3E-15	-4,02E-15		-1906,95	-10842,5	Δ q1
2	0,16302	-0,3804	0,0055	-0,013	-0,4369	3,036595	3,036595	0,835183	0,835183	2,201412		-808,205	-19156,9	Δ q2
3	-0,2025	0,47254	-0,0068	0,016	0,5428	4,489581	4,489581	-1,0375	-1,0375	5,5270806	*	-4045,3	-22016,5	Δ q3
4	-0,137	0,31961	-0,2945	0,6872	0,36713	3,036595	3,036595	0,835183	0,835183	2,201412	=	-2810,46	-22421,6	Δ q4
5	0,01596	-0,0372	0,0343	-0,08	-0,0428	-0,35371	-0,35371	5,204774	5,204774	-5,558481		-2735,84	-23388	Δ q5
2	0,02451	-0,0572	0,00083	-0,002	-0,0657	-0,54342	0,456584	0,125578	0,125578	0,3310052		-2615,2	-69,985	Δ H2
3	-0,003	0,00702	-0,0065	0,0151	0,00807	0,066711	0,066711	-0,98165	0,018348	1,0483631		-2287,5	2123,622	Δ h3
4	0	0	8,9E-16	2E-15	1,1E-16	1,18E-14	1,15E-14	8,88E-15	8,44E-15	1		1	1	Δ H4

In addition to nodal demand framework analysis, one advantage of the introduction of Inverse matrix method associated with the iterative procedure is the easy performance of obtaining directly unmeasured pipe flows and nodal pressures with no need to calculate the forward computation again. A point of departure for the feasibility of the model validation is the intrinsic quickness of the inversion of the model, that can solve the nodal demand aggregation in four iterations as seen in Table 19 and Fig. 12. At this table, it is shown the corrections of grouped demand after iterations. These corrections of the grouped demand residuals are originated by the difference between the absolute sum of the *Δ*M in each iteration according to the Eq. (17).

The efficiency and the robustness of the present model are evident from several successful runs made in the Classical Inverse method and Inverse model.

Table 19. Grouped Demand residuals

k	$\Delta \mathbf{G}_1^{(k)}$	$\Delta \mathbf{G}_2^{(k)}$
1	7169.710	1427.623
2	1.809 E-10	1.042 E-10
3	5.911 E-12	2.387 E-11
4	2.728 E-12	2.387 E-12

The intention of given initial nodal Heads and pipe flows is to calibrate the model based on nodal demands within a few three or four iterations. It's explained because water distribution systems integrate soft and weak nonlinear systems. After the needed corrections, calibration data analysis was performed.

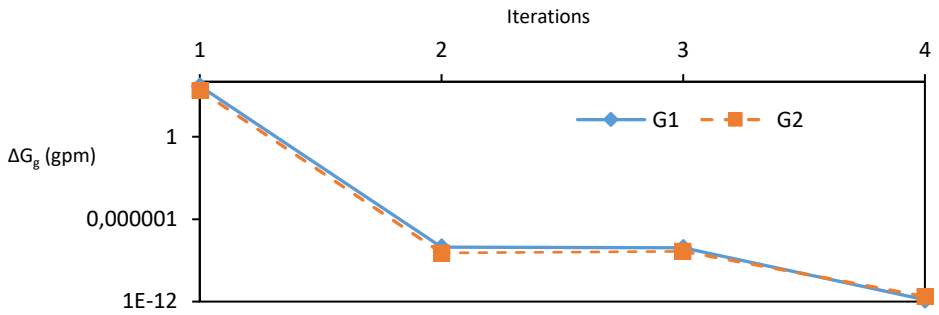


Fig. 12. Convergence of nodal demands for the solution based on the Inverse model equations

The consumption of water in time and space of the supplied area as the initial pipes roughness was considered the same as the Classic Inverse method to get similar results on the calibration process. These results have the demand allocation matrix behind the nodal demand calculation, where Eq. (13) is responsible to filter the grouped nodal demands of each iteration and can be calculated by

$$\mathbf{M}_{uM} = \mathbf{G}_d \mathbf{M}_g \,, \tag{23}$$

and

$$\mathbf{M}_{uM} = \begin{bmatrix} 0.7 & 0 \\ 0.3 & 0 \\ 0 & 0.7 \\ 0 & 0.3 \end{bmatrix} \cdot \begin{bmatrix} 1150.23 \\ 2699.28 \end{bmatrix} = \begin{bmatrix} 805.16 \\ 345.07 \\ 1889.50 \\ 809.79 \end{bmatrix}, \tag{24}$$

where shows the grouped nodal demands calibrated to the real values at fourth iteration, and then it is possible to see the individual nodal demand dividing the first two cells in the first group and the other two cells as the second group. In Table 20 all the nodal demands at each iteration can be seen. The results originated from the disintegration of the grouped nodal demands can be estimated as showed in Table 21.

Table 20. Nodal demands in iteration process

Iterations	Node 1 (gpm)	Node 2 (gpm)	Node 3 (gpm)	Node 4 (gpm)
0	803.88	344.52	1910.39	818.74
1	4.86	-2.08	2157.89	924.81
2	6327.08	2711.61	-3560.22	-1525.81
3	2911.61	1247.83	-671.09	-287.61
4	805.16	345.07	1889.50	809.79

Table 21. Nodal demand results of the Inverse Model

Nodes	$\mathbf{M}_{uM}$ (gpm)
1	805.16
2	345.07
3	1889.50
4	809.79

In this regard, by using two different approaches to solve the exact same problem it is taken into account that there isn't an uncertainty analysis of the data. After looking through the results of both methods it is possible to calculate the difference between them. The percentage difference calculated is 5,88 %.

4.3 Transient

The second regime was created from a point of view of making a real simulation of the network for the period of 24 h. In this context, this scenario was created with the assistance of Epanet. This hydraulic solver is a powerful tool for simulation and modeling network. The Epanet is responsible for running the true pipe flows and nodal heads to be included in the calibration methodology. In this way is possible to validate the successive nodal demands of the model calibration over time. The network and all the parameters verify the same conditions as exemplified in the predefined network. The context of transient methodology in terms of reproducing the algorithm planned is a little different of the steady-state and can be assumed as the main divergence of both scenarios. In the transient context, the Inverse model is the only method used for the results construction.

5. Results and validation

5.1 Steady-state

In Tables 22 the flows and heads result from the iteration process of the Classical Inverse model are exposed.
The tables below show the role of the parameters implanted in the calibration process. Table 22 shows the importance of knowing values of the last iteration of the Classical Inverse model on which are the initial values of the Inverse model, shown in Table 23. The other parameters of the method results are exposed on Appendix A. About the Inverse model, the initial values are presented in Table 23.

Table 22. Final values of the fourth iteration of Classical Inverse method

Index	**H** (ft) | **q** (gpm)
H_2	1721.43
H_3	1253.34
H_4	139.13
q_1	6917.15
q_2	1600.98
q_3	782.67
q_4	-1071.79
q_5	-1409.54

For nodal demand calibration is important to remind that the pattern for each node is based on the demand allocation matrix $\mathbf{G}_d$, as mentioned in the Inverse model procedure. All the model doesn't suffer any alteration in terms of the pipe parameters and structural framework. In Table 24, the grouped nodes could be seen on the pairs of cells called the $\mathbf{M}_{g,1}$ and $\mathbf{M}_{g,2}$ with respect to Eq. (13) and Eq. (17). The desegregated nodal demands are shown in Table 25.

Table 23. Initial values of the Inverse model

Index	$\mathbf{q}_{uq}$ (gpm)		$\mathbf{H}_{uH}$ (ft)
q_1	6917.15		-
q_2	1600.98		-
q_3	782.67	H_2	1721.43
q_4	-1071.79	H_3	1253.34
q_5	-1409.54	H_4	139.13

The consumption of water in time of the supplied area has been determined as possible due to his importance for the convergence and stability of the algorithm. The true values are called as the values originated from the Direct method and are shown in Table 24.

Table 24. Final values of the fourth iteration of Inverse model

Index	$\mathbf{M}_g$ | $\mathbf{q}_{uq}$ | $\mathbf{H}_{uH}$
$M_{g,1}$	1150.23
$M_{g,2}$	2699.28
q_1	7672.36
q_2	1148.44
q_3	-3090.4
q_4	-2346.6
q_5	-427.39
H_2	3714.66
H_3	2311.82
H_4	1584.26

Table 25. Nodal demands result for the fourth iteration of the Inverse model

Nodes	$\mathbf{M}_{uM}$ (gpm)
1	805.16
2	345.07
3	1889.50
4	809.79

It is important to keep in mind that the flows and heads don't need to be the same at the last iterations of the model due the differences of the methods application. The initial, true and calibrated values for nodal demands are executed and could be compared, as seen in Table 26. After this evidence, it is expected the model convergence of the two methods due to the almost non-existent variation of the nodal demands. After this, it is possible to see the convergence of the two methods created, meaning the purpose of the Steady-state modeling.

Table 26. Initial, True and Calibrated nodal demands values for the Steady-State Case

Index	Initial (gpm)	True (gpm)	Calibrated (gpm)
Node 1	818.00	818.31	805.16
Node 2	330.39	337.75	345.07
Node 3	1855.26	1854.46	1889.50
Node 4	873.86	810.60	809.79

The comparison of the results of Table 21 and Table 25 indicates the producing of lower errors mainly on the node with more flow, not falling far short of the real value.

5.2 Transient

Considering the base consumption of each junction, the consumption values for each hour of the day are calculated by the Epanet program, considering the consumption pattern. The standards are usually set by the Epanet user even before any simulation.

In the same way as the Steady-state, it is important to know the final values of the nodal demands group showing in detail all the results of pipe flow, nodal head, and grouped nodal demands. As mention before the grouped nodal demands continues to be run with the influence of the demand allocation matrix, where for the two initials nodes are considered 0.7 and 0.3, respectively, as the ratio of consumption conversion shown in Eq. (15). For that reason, the nodal demand 1 and 2 is the grouped nodal demand, $M_{g,1}$ as presented in Table 27.

The initial values of nodal demands for the Transient regime are collected from the Inverse model for nodal demand where they have been considered as constant during the time.

Table 27. Final values of the fourth iteration at 06:00 hours

index	$\mathbf{M}_g \mid \mathbf{q}_{uq} \mid \mathbf{H}_{uH}$ (gpm)
$M_{g,1}$	9803.825
$M_{g,2}$	5710.340
q_1	7971.018
q_2	24.413
q_3	-3247.134
q_4	-2407.897
q_5	-265.927
H_2	16265.102
H_3	14651.977
H_4	16366.910

Table 28. Nodal demands result for the fourth iteration at 06:00 h

Nodes	$\mathbf{M}_{uM}$ (gpm)
1	6862.68
2	2941.15
3	3997.24
4	1713.10

These initial parameters are represented in Table 29. Due to the history and acceptance of Epanet, the results from its simulation were considered the correct values with a free error. The simulation of Epanet values was constant at each hour and worked by solving the Hazen Williams equations. All the additional features were implemented externally to the Epanet and the Excel was responsible to integrate all this data provided from the Epanet. The parameters established in the Epanet were all faithfully represented as the Inverse model.

These parameters are exemplified as the case of Hazen Williams coefficient, tank diameter, nodes elevation and flow units. The Epanet files reproduce not only graphs and tables related to nodal demands but also other parameters as pressures in the nodes, heads and base demand.

The initial values presented in Table 29 are retrieved from the model inverse sheet. The process takes these values to compare with the calibrated and true nodal demand.

Table 29. Initial nodal demands for the Transient Case (24 hours)

Hour (t)	Node 1	Node 2	Node 3	Node 4
$\mathbf{M}_i$ (gpm)	818.38	329.49	1855.27	873.60

The true and calibrated demands for each node in 24 h are shown in Tables 30 - 33. The simulation of the simple network in Epanet improved the understanding of the hydraulic simulation capabilities [20].

Table 30. Nodal demand results for node 1 of Transient Case (gpm)

Hours	Initial	True	Calibrated
0	818.38	801.00	805.16
1	818.38	2848.00	1871.08
2	818.38	7654.00	5064.09
3	818.38	4984.00	6348.85
4	818.38	6230.00	6818.77
5	818.38	5340.00	7105.10
6	818.38	8010.00	6862.68
7	818.38	6052.00	7094.81
8	818.38	7476.00	6723.64
9	818.38	7832.00	6895.81
10	818.38	8099.00	7434.96
11	818.38	7743.00	6832.51
12	818.38	6141.00	7260.17
13	818.38	6176.60	7163.76
14	818.38	5518.00	7081.91
15	818.38	7689.60	6949.12
16	818.38	7654.00	6838.24
17	818.38	6319.00	7170.58
18	818.38	6336.80	7013.15
19	818.38	6790.70	6814.45
20	818.38	8250.30	8789.51
21	818.38	12638.00	11999.60
22	818.38	7921.00	7115.45
23	818.38	8277.00	7388.65

Table 31. Nodal demand results for node 2 of Transient Case (gpm)

Hours	Initial	True	Calibrated
0	329.49	297.00	345.07
1	329.49	1056.00	801.89
2	329.49	2838.00	2170.32
3	329.49	1848.00	2720.93
4	329.49	2310.00	2922.33
5	329.49	1980.00	3045.04
6	329.49	2970.00	2941.15
7	329.49	2244.00	3040.63
8	329.49	2772.00	2881.56
9	329.49	2904.00	2955.35
10	329.49	3003.00	3186.41
11	329.49	2871.00	2928.22
12	329.49	2277.00	3111.50
13	329.49	2290.20	3070.18
14	329.49	2046.00	3035.11
15	329.49	2851.20	2978.20
16	329.49	2838.00	2930.68
17	329.49	2343.00	3073.11
18	329.49	2349.60	3005.64
19	329.49	2517.90	2920.48
20	329.49	3059.10	3766.93
21	329.49	4686.00	5142.68
22	329.49	2937.00	3049.48
23	329.49	3069.00	3166.57

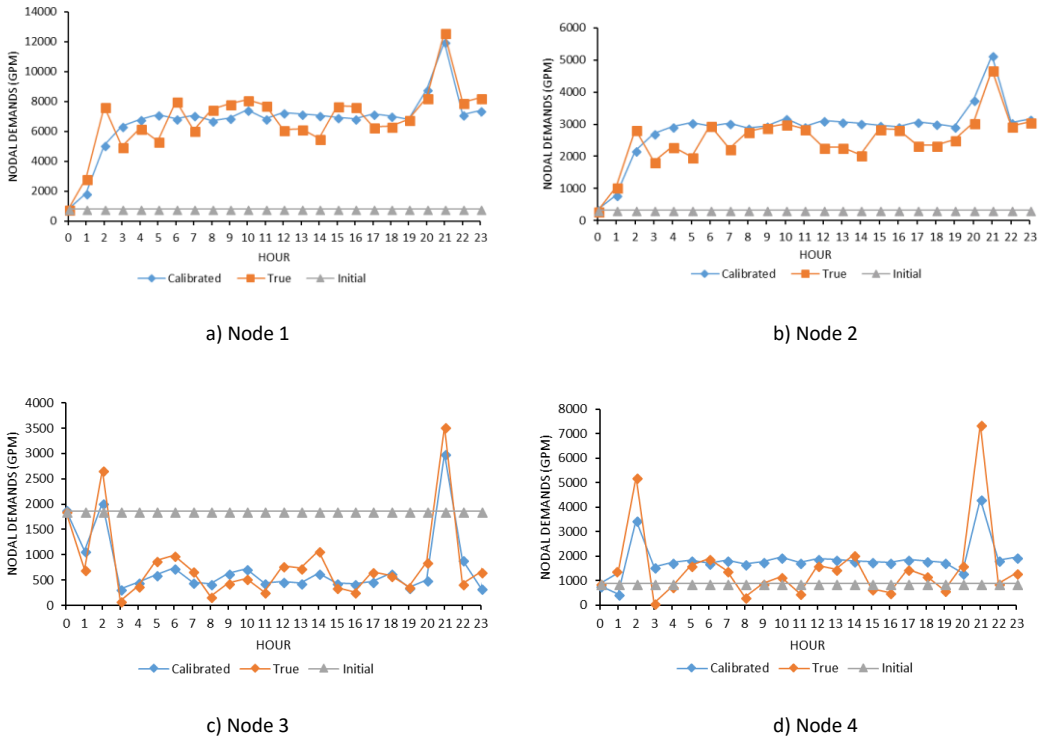
Table 32. Nodal demand results for node 3 of Transient Case (gpm)

Hours	Initial	True	Calibrated
0	1855.27	1858.71	1889.46
1	1855.27	704.90	1079.70
2	1855.27	2671.20	2025.25
3	1855.27	85.33	321.84
4	1855.27	389.55	463.90
5	1855.27	890.40	608.55
6	1855.27	983.15	743.21
7	1855.27	686.35	460.73
8	1855.27	185.50	426.84
9	1855.27	445.20	641.91
10	1855.27	537.95	725.34
11	1855.27	259.70	438.78
12	1855.27	779.10	479.21
13	1855.27	742.00	438.13
14	1855.27	1075.90	639.49
15	1855.27	352.45	437.34
16	1855.27	259.70	423.78
17	1855.27	667.80	478.18
18	1855.27	593.60	642.53
19	1855.27	371.00	348.72
20	1855.27	853.30	497.62
21	1855.27	3524.50	3000.97
22	1855.27	426.65	900.26
23	1855.27	667.80	340.72

Table 33. Nodal demand results for node 4 of Transient Case (gpm)

Hours	Initial	True	Calibrated
0	873.60	884.40	809.79
1	873.60	1326.60	462.73
2	873.60	5232.70	3467.96
3	873.60	73.70	1574.03
4	873.60	759.11	1743.51
5	873.60	1621.40	1847.83
6	873.60	1916.20	1713.10
7	873.60	1415.04	1844.21
8	873.60	346.39	1701.60
9	873.60	891.77	1775.39
10	873.60	1179.20	1979.15
11	873.60	486.42	1741.99
12	873.60	1621.40	1908.45
13	873.60	1474.00	1870.57
14	873.60	2063.60	1824.59
15	873.60	685.41	1787.34
16	873.60	515.90	1744.71
17	873.60	1474.00	1873.21
18	873.60	1208.68	1812.77
19	873.60	604.34	1739.83
20	873.60	1621.40	1307.39
21	873.60	7370.00	4320.34
22	873.60	884.40	1851.42
23	873.60	1326.60	1953.01

The corresponding Epanet input file that provide the nodal demand for each hour can be found more detailed in Appendix B. The results showed in the last tables can be easily compared in the Fig. 13, showing there is a good agreement between the calibrated and the true nodal demands for a period of 24 h. It is believed that good fitness of observation values means a good calibration. The results from this transient case could be validated with the Epanet files. For each node, it is possible to match the nodal demands and compares the feasibility of the methodology created.



a) Node 1

b) Node 2

c) Node 3

d) Node 4

Fig. 13. Calibration results of Transient Case study

The comparison of nodal demand in a way of individualized nodes sometimes is meaningless. In the practice, a good calibration algorithm or monitor network has the purpose to improve the estimation of nodal demand. In this case, the real nodal demand of every node respects the flow mass conservation. Due to the engineering knowledge, it can be assumed the feasibility of the calibration results. This approach was very important through the analysis and validation of the proposed framework. For the Transient Case, there are some divergences of the nodal demand results due to the application of the Inverse model. It must be remembered that, for a distinct case, for example, with the addition of a pump or even a distinct system, the results obtained by Epanet may overestimate the real results. In this case, the introduction of properties that play a differentiating role in the hydraulics behavior of the network can be analyzed as a different problem with a spatial and temporal nature of the water use [21].

In addition, to further evaluate the calibration results, Table 34 gives the estimation of all unknowns in the first hour, including the grouped nodal demand, pipe flow, and nodal heads. With the creation of Transient Case appears the question of monitorization the water level of the tank. In this case, it's considered a tank for the calculation and results, which is composed for an initial tank water level, the time interval between contiguous measurements, the sectional area and with the final level in each time step.

Table 34. Estimates of all unknowns in the second hour

All unknowns	Estimates
M_1	2672.96 (gpm)
M_2	1542.44 (gpm)
Pipe 1	5008.06 (gpm)
Pipe 2	-683.39 (gpm)
Pipe 3	-1815.69 (gpm)
Pipe 4	-2226.82 (gpm)
Pipe 5	-3002.38 (gpm)
Node 2	7637.73 (ft)
Node 3	6573.68 (ft)
Node 4	7086.62 (ft)

By repeating the updated water level in each iteration of the transient Case it is possible to know the tank's water level and outflow in each hour of the day. The results are shown in Table 35 where it´s possible to see the behavior of the network. It is normal that with time the level of water is rising and stabilizes at a maximum level of the tank. The application of the Eq. (24) performed in the Excel, is used to disintegrate the nodal demands.

Table 35. Tanks' Water level for 24 h (Inverse model)

Hour	Tank water level (ft)	Hour	Tank water level (ft)
0	2529.00	12	19180.75
1	9604.38	13	19697.32
2	24124.99	14	17114.66
3	12572.02	15	24903.51
4	19904.62	16	25110.26
5	17838.16	17	19738.63
6	25067.50	18	20771.78
7	19800.72	19	21805.12
8	23003.32	20	25929.81
9	24242.82	21	38275.23
10	27134.04	22	24583.06
11	24965.65	23	26038.94

Table 36. Tank's Water level and outflow at 03:00 h

H^{i+1}_{tank} (t) (ft)	H^{0}_{tank} (ft)	q_1(t) (gpm)	Δt (min)	A (ft²)	q_{in} (gpm)
12572.02	2529	7700.56	60	5808.8	980000

The Epanet brings a comparative advantage to know the draining and filling behavior of the tank. The Fig. 14 shows the data correlation of the Water height of tank used by Epanet and Inverse model for the Transient case.

Table 36 shows the parameters linked to the water tank evolution at 03:00 h. At this point, the q_1 is provided from the fourth and last iteration of the Inverse model applicated in this predefined Case study. By repeating this procedure every hour, the tank's water level can be updated and determined by the influence of the outflow of pipe 1.

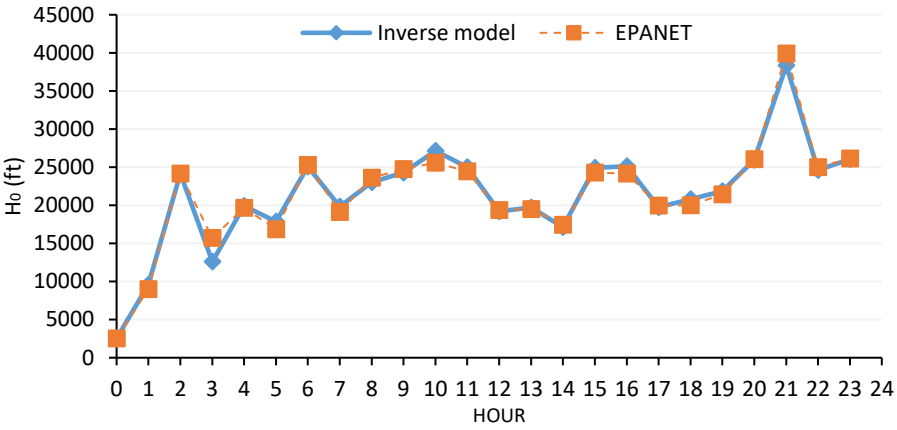


Fig. 14. Water height of tank comparison between Epanet and Inverse model

Although the Inverse model method is more efficient at achieving the results in a shorter processing time due to the iterative steps, it can be said that both methods used in this dissertation show a good applicability of the calibration problem.

6. Summary and conclusions

This work shows two methods for nodal demand calibration.
It is explained how these two different approaches to solving the same engineering problem work in terms of methodology advantages and performances. The Classical Inverse method it's originated from Todini view and combines the Gradient method with the optimization process.
The Inverse model for nodal calibration can take advantage of the gradient method, by using a hybrid algorithm to separate the known and unknown variables. In this case, there isn't a direct optimization of the results due to the Gauss-Newton iteration method. This is the major advantage when the model is determined, thus allowing the entry of the unknown nodal demand, pipe flow, and nodal head into the model resolution. To evaluate the calibration results a Case study for the Steady-state regime and for the Transient regime was created. The first regime, Steady-state, is created to validate the numerical methods used for nodal demand calibration. In that case, it is used the data provided by the Classic Inverse method. This information is then incorporated in the Inverse model where it demonstrates that the feasibility and effectiveness of the results is ensured. The updated variables from one method to the other are the nodal heads and pipe flows. The Transient Case study uses a different perspective of the calibration method once the initial data is created by the Epanet software. The generation by Epanet of nodal heads and pipe flow is used for the Inverse model as input parameters to solve the problem. As the Inverse model run the unknown parameters it is possible to know the height of the water tank in each time step of the network and analyse the water variation over time. Epanet is based on Todini's formulation of hydraulic equations known as the gradient method.
The calibration of any water distribution systems hydraulic model is usually a very complex task. Rather than using some trial-and-error approach, an optimization type procedure should be used to solve it. Still, it is not reasonable to expect that, in general, the calibration problem will be solved using a completely automated procedure [12].
Calibration of the water distribution network nowadays is an ill-posed problem as viewed in the above sentence and in practice, the problem solution can be a consequence of inadequate quantity and quality of observation information [22].
By using the proposed methods and their validation, the results can confirm the potential for practical application of nodal demand calibration.
For future work development, several issues could be solved, as the estimation of several parameters errors and an uncertainty analysis. Another way of completing this type of work is extending the proposed framework to include pipe roughness or another hydraulic parameter to acquire more feasible and robust results. Considering the presented methodology, it would be interesting and useful test the developed model using data measured in a real network. Another possible way to develop the present work is the implementation of the methodology for a generic problem with no need to use a Case-study.

References

[1] WWAP (United Nations World Water Assessment Programme). Wastewater. The Untapped Resource, The United Nations World Water Development Report. (2017) Retrieved November 23, 2017, from Unesco: http://unesdoc.unesco.org/images/0024/002471/247153e.pdf.

[2] Coelho, B. D. C. Energy Efficiency of Water Supply Systems Using Optimization Techniques. Ph.D. Dissertation. [Aveiro]: Universidade de Aveiro. (2016).

[3] Walski, T. M. Technique for Calibrating Network Models, *Journal of Water Resources Planning and Management*, 109, 4 (1983), 360-372.

[4] Sanz, G. and Pérez, R. Demand pattern calibration in water distribution networks, *Procedia Engineering*, 70, (2014), 1495-1504.

[5] Kapelan Z. Calibration of Water Distribution Networks. Ph.D. Dissertation. [Exeter]: University of Exeter. (Feb, 2002), 78-136.

[6] Preis, A. et al. Real-Time Hydraulic Modelling of a Water Distribution System in Singapore, (2012) *14th Water Distribution Systems Analysis Conference (WDSA 2012), Adelaide, Australia, September 24-27*, (2012), 3-6

[7] Muranho, J., Ferreira, A., Sousa, J., Gomes, A., Sá Marques, A. Convergence issues in the EPANET solver. Procedia Engineering, 119,1 (2015), 700-709.

[8] Todini, E., and Pilati, S. A gradient method for the analysis of pipe networks. In: Coulbeck, B. Orr, C. H., editors. *Proceedings of the International Conference and Computer Applications for Water Supply and Distribution*, 1 - Systems Analysis and Simulation (May 1987).

[9] López-Ibáñez, M. Operational optimization of water distribution networks. Ph.D. Dissertation. [Edinburgh]: Edinburgh Napier University. (Nov, 2009), 34-63

[10] Rao, Z. and Salomons, E. Development of a real-time, near optimal control process for water distribution networks. *Journal of Hydroinformatics*, 9, 1 (2007), 25-37.

[11] Sonaje, N. P. and Joshi, M. G. A Review of Modeling and Application of Water Distribution Networks (Wdn) Softwares. *International Journal of Technical Research and Applications*, 3, 5 (2015), 174-178.

[12] Ormsbee, L. E. Implicit network calibration, 115, 2 (1989), 243–257.

[13] Laucelli, D. et al. Calibration of Water Distribution System Using Topological Analysis. *Water Distribution Systems Analysis 2010*, (i) (2010) 1664–1681. Retrieved December 2, 2017 from http://cedb.asce.org/cgi/WWWdisplay.cgi?284733

[14] Bhave, P. R. Calibrating Water Distribution Network Models. *Journal of Environmental Engineering* 114, 1 (1988), 120-136.

[15] Sunela, M. Real-Time Control Optimization of Water Distribution System with Storage. Ph.D. Dissertation. [Tallin]: Tallin University of Technology. (Oct, 2017), 123-136

[16] Lingireddy, S., and Ormsbee, L. E. Hydraulic network calibration using genetic optimization. *Civil Engineering and Environmental Systems*, 19, 1 (2002), 13-39.

[17] Orłowska, M. The optimization of calibration of the computer model describing the flows in the water distribution network. *E3S Web of Conferences,* 17, 69 (2017)

[18] Kun, D., Tian-Yu, L., Jun-Hui, W., and Jin-Song, G. Inversion Model of Water Distribution Systems for Nodal Demand Calibration. *Journal of Water Resources Planning and Management*, 141, 9 (2015), 1-12.

[19] Newbold, J. R. Comparison and Simulation of a Water Distribution Network in EPANET and a New Generic Graph Trace Analysis Based Model. Master's Thesis. [Virginia]: Virginia Polytechnic Institute and State University. (Jan, 2009), 46-49

[20] Rossman, L.A. *EPANET 2 Users Manual*. Water Supply and Water Resources Division National Risk Management Research Laboratory. *Cincinnati US Environmental Protection Agency National Risk Management Research Laboratory*,38 (2000) 123-126.
[21] Cheng, W., He, Zhiguo. Calibration of Nodal Demand in Water Distribution Systems. *Journal of Water Resources Planning and Management*, 137, 1 (2011), 31-40
[22] Tabesh, M., Jamasb, M., and Moeini, R. Calibration of water distribution hydraulic models: A comparison between pressure dependent and demand driven analyses. *Urban Water Journal*,8 ,2 (2011) 93-102.

Appendices

A. Steady-state Operations and Results

Classical Inverse method with the implementation of the Direct method:

Tube	node i	node j	C-factor	d(inch)	L(ft)	R (H.Wil)
1	0	1	90	7,874	1640,4	0,000181
2	1	2	90	7,874	1640,4	0,000181
3	2	4	90	7,874	1640,4	0,000181
4	4	3	90	7,874	1640,4	0,000181
5	3	1	90	7,874	1640,4	0,000181

Hazen-williams	
e=	1,85
Ku=	10,44

A	tube 1	tube 2	tube 3	tube 4	tube 5
node 2	0	1	-1	0	0
node 3	0	0	0	1	-1
node 4	0	0	1	-1	0

A^(T)		
0	0	0
1	0	0
-1	0	1
0	1	-1
0	-1	0

A				
0	1	-1	0	0
0	0	0	1	-1
0	0	1	-1	0

B				
2,69217	0	0	0	0
0	8,163337946	0	0	0
0	0	7,134395	0	0
0	0	0	3,2993567	0
0	0	0	0	23,17362943

AB				
0	8,16334	-7,134395	0	0
0	0	0	3,299357	-23,174
0	0	7,1343948	-3,29936	0

A^(T)		
0	0	0
1	0	0
-1	0	1
0	1	-1
0	-1	0

BA^(T)		
0	0	0
8,163337946	0	0
-7,134394797	0	7,1343948
0	3,29936	-3,299357
0	-23,174	0

ABA^(T)		
15,2977	0	-7,13439
0	26,472986	-3,29936
-7,1344	-3,2993567	10,43375

[ABA^(T)]^-1		
0,097854775	0,008681	0,0696564
0,008681351	0,040094	0,0186148
0,069656391	0,018615	0,1493588

JACOBIAN INVERSE (nxm)							
0,000	0,000	0,000	2,692	0,000	0,000	0,000	0,000
0,799	0,071	0,569	0,000	1,642	1,642	1,642	1,642
-0,201	0,071	0,569	0,000	1,642	1,642	1,642	1,642
-0,201	0,071	-0,431	0,000	1,642	1,642	1,642	1,642
-0,201	-0,929	-0,431	0,000	1,642	1,642	1,642	1,642
-0,098	-0,009	-0,070	0,000	0,799	-0,201	-0,201	-0,201
-0,009	-0,040	-0,019	0,000	0,071	0,071	0,071	-0,929
-0,070	-0,019	-0,149	0,000	0,569	0,569	-0,431	-0,431

H0 | fixed head + heads				
1	H0=	2300	A10H0=	2300
0				0
0				0
0				0
0				0

Iterations 1 - 2:

	Jacobian (nxm)									
iteration:1	index	H| q	1	2	3	4	5	2	3	4
	2	2790	0	1	-1	0	0	0	0	0
	3	1800	0	0	0	1	-1	0	0	0
	4	1400	0	0	1	-1	0	0	0	0
	1	3821,1324	0,371448	0	0	0	0	0	0	0
	2	1036,1175	0	0,1225	0	0	0	1	0	0
	3	1214,0737	0	0	0,1401661	0	0	-1	0	1
	4	3008	0	0	0	0,303089	0	0	1	-1
	5	303,61223	0	0	0	0	0,04315	0	-1	0

Inverse Jacobian			Validation				
0	0	0	2,69217	0	0	0	0
0,79882	0,0708688	0,568629	0	1,6422873	1,642287	1,64229	1,642287305
-0,2012	0,0708688	0,568629	0	1,6422873	1,642287	1,64229	1,642287305
-0,2012	0,0708688	-0,43137	0	1,6422873	1,642287	1,64229	1,642287305
-0,2012	-0,929131	-0,43137	0	1,6422873	1,642287	1,64229	1,642287305
-0,0979	-0,008681	-0,06966	0	0,7988216	-0,20118	-0,2012	-0,2011784
-0,0087	-0,040094	-0,01861	0	0,0708688	0,070869	0,07087	-0,9291312
-0,0697	-0,018615	-0,14936	0	0,5686287	0,568629	-0,4314	-0,43137134

$\begin{bmatrix} \Delta\mathbf{q}^{(k)} \\ \Delta\mathbf{H}^{(k)} \end{bmatrix}$

$\Delta\mathbf{M}^{(k)}$	996,27	0	-8257,5	-8257,46
	-2366,6	2702,699	-1084,7	1618,00
	3648,39	1706,429	-1084,7	621,73
	-3067,2	-1941,96	-1084,7	-3026,66
	-68,607	424,6762	-1084,7	-660,02
$\Delta\mathbf{h}^{(k)}$	-91,985	-331,078	64,2673	-266,81
	-492,81	18,32584	-39,726	-21,40
	-7,082	-570,261	124,321	-445,94

	jacobiano(nxm)									
iteration:2	index	H| q	1	2	3	4	5	2	3	4
	2	2523,1896	0	1	-1	0	0	0	0	0
	3	1778,6003	0	0	0	1	-1	0	0	0
	4	954,05949	0	0	1	-1	0	0	0	0
	1	-4436,331	0,421701	0	0	0	0	0	0	0
	2	2654,1171	0	0,2725	0	0	0	1	0	0
	3	1835,8028	0	0	0,1991988	0	0	-1	0	1
	4	-18,65753	0	0	0	0,00403	0	0	1	-1
	5	-356,411	0	0	0	0	0,04945	0	-1	0

Inverse Jacobian							
0	0	0	2,371347	0	0	0	0
0,481132	0,09416	0,1018365	0	1,9041	1,904103607	1,904104	1,9041036
-0,51887	0,09416	0,1018365	0	1,9041	1,904103607	1,904104	1,9041036
-0,51887	0,09416	-0,8981635	0	1,9041	1,904103607	1,904104	1,9041036
-0,51887	-0,9058	-0,8981635	0	1,9041	1,904103607	1,904104	1,9041036
-0,13111	-0,0257	-0,0277504	0	0,48113	-0,5188684	-0,518868	-0,518868
-0,02566	-0,0448	-0,0444169	0	0,09416	0,094163705	0,094164	-0,905836
-0,02775	-0,0444	-0,0480361	0	0,10184	0,101836458	-0,898164	-0,898164

$\begin{bmatrix} \Delta\mathbf{q}^{(k)} \\ \Delta\mathbf{H}^{(k)} \end{bmatrix}$

$\Delta\mathbf{M}^{(k)}$	-3E-13	0	-3056,0809	-3056,08
	0	-2E-13	-726,22103	-726,221
	-2E-13	1,5E-13	-726,22103	-726,221
	-1288,8	3,8E-13	-726,22103	-726,221
	-390,94	3,8E-13	-726,22103	-726,221
$\Delta\mathbf{h}^{(k)}$	-0,0214	5,1E-14	-193,04911	-193,049
	0,04064	1,9E-14	-45,441189	-45,4412
	9,52735	2E-14	-48,408199	-48,4082

9713,512741

Iterations 3 - 4:

	jacobiano(nxm)									
iteration:3	index	H| q	1	2	3	4	5	2	3	4
	2	2330,1405	0	1	-1	0	0	0	0	0
	3	1733,1591	0	0	0	1	-1	0	0	0
	4	905,65129	0	0	1	-1	0	0	0	0
	1	-7492,412	0,658359	0	0	0	0	0	0	0
	2	1927,8961	0	0,20766	0	0	0	1	0	0
	3	1109,5818	0	0	0,1298434	0	0	-1	0	1
	4	-744,8786	0	0	0	0,092535	0	0	1	-1
	5	-1082,632	0	0	0	0	0,12716	0	-1	0

Inverse Jacobian							
0	0	0	1,518927	0	0	0	0
0,627311	0,22821	0,3942821	0	1,79469	1,794694424	1,794694	1,7946944
-0,37269	0,22821	0,3942821	0	1,79469	1,794694424	1,794694	1,7946944
-0,37269	0,22821	-0,6057179	0	1,79469	1,794694424	1,794694	1,7946944
-0,37269	-0,7718	-0,6057179	0	1,79469	1,794694424	1,794694	1,7946944
-0,13027	-0,0474	-0,0818771	0	0,62731	-0,37268865	-0,372689	-0,372689
-0,04739	-0,0981	-0,0770218	0	0,22821	0,228209551	0,22821	-0,77179
-0,08188	-0,077	-0,1330721	0	0,39428	0,394282059	-0,605718	-0,605718

$$\begin{bmatrix} \Delta\mathbf{q}^{(k)} \\ \Delta\mathbf{H}^{(k)} \end{bmatrix}$$

			0	556,4193	556,4193
$\Delta\mathbf{M}^{(k)}$	-1E-13		1,8E-14	-327,72932	-327,729
	0		1,3E-13	-327,72932	-327,729
	2,3E-13		-1E-13	-327,72932	-327,729
	366,324		-1E-13	-327,72932	-327,729
	-216,41		-4E-15	-148,34842	-148,348
$\Delta\mathbf{h}^{(k)}$	-77,877		-1E-14	-116,08699	-116,087
	37,2581	2009,575367	-2E-14	-183,67163	-183,672
	74,4136				

	jacobiano(nxm)									
iteration:4	index	H| q	1	2	3	4	5	2	3	4
	2	2181,7921	0	1	-1	0	0	0	0	0
	3	1617,0721	0	0	0	1	-1	0	0	0
	4	721,97967	0	0	1	-1	0	0	0	0
	1	-6935,993	0,616562	0	0	0	0	0	0	0
	2	1600,1668	0	0,17725	0	0	0	1	0	0
	3	781,85249	0	0	0,0964252	0	0	-1	0	1
	4	-1072,608	0	0	0	0,126156	0	0	1	-1
	5	-1410,361	0	0	0	0	0,15921	0	-1	0

Inverse Jacobian							
0	0	0	1,621896	0	0	0	0
0,682944	0,28479	0,5104591	0	1,7888	1,788795877	1,788796	1,7887959
-0,31706	0,28479	0,5104591	0	1,7888	1,788795877	1,788796	1,7887959
-0,31706	0,28479	-0,4895409	0	1,7888	1,788795877	1,788796	1,7887959
-0,31706	-0,7152	-0,4895409	0	1,7888	1,788795877	1,788796	1,7887959
-0,12105	-0,0505	-0,0904765	0	0,68294	-0,31705588	-0,317056	-0,317056
-0,05048	-0,1139	-0,0779389	0	0,28479	0,284790952	0,284791	-0,715209
-0,09048	-0,0779	-0,1396977	0	0,51046	0,510459064	-0,489541	-0,489541

$$\begin{bmatrix} \Delta\mathbf{q}^{(k)} \\ \Delta\mathbf{H}^{(k)} \end{bmatrix}$$

			0	18,823605	18,82361
$\Delta\mathbf{M}^{(k)}$	0		-8E-14	0,8169628	0,816963
	1,1E-13		-8E-14	0,8169628	0,816963
	-2E-13		1,4E-13	0,8169628	0,816963
	11,6059		3E-14	0,8169628	0,816963
	-153,31		1,5E-14	-153,45412	-153,454
$\Delta\mathbf{h}^{(k)}$	-40,752		4,8E-15	-121,24348	-121,243
	73,144	537,7846981	2,3E-14	-194,28441	-194,284
	121,374				

Iterations 5 - 6:

	jacobiano(nxm)									
iteration:5	index	H| q	1	2	3	4	5	2	3	4
	2	2028,338	0	1	-1	0	0	0	0	0
	3	1495,8287	0	0	0	1	-1	0	0	0
	4	527,69525	0	0	1	-1	0	0	0	0
	1	-6917,169	0,61514	0	0	0	0	0	0	0
	2	1600,9837	0	0,1773223	0	0	0	1	0	0
	3	782,66945	0	0	0,0965109	0	0	-1	0	1
	4	-1071,791	0	0	0	0,126075	0	0	1	-1
	5	-1409,544	0	0	0	0	0,15913	0	-1	0

Inverse Jacobian							
0	0	0	1,625647	0	0	0	0
0,682808	0,2846495	0,5101705	0	1,78879	1,788787947	1,788788	1,7887879
-0,31719	0,2846495	0,5101705	0	1,78879	1,788787947	1,788788	1,7887879
-0,31719	0,2846495	-0,4898295	0	1,78879	1,788787947	1,788788	1,7887879
-0,31719	-0,7153505	-0,4898295	0	1,78879	1,788787947	1,788788	1,7887879
-0,12108	-0,0504747	-0,0904646	0	0,68281	-0,31719206	-0,317192	-0,317192
-0,05047	-0,1138336	-0,0779465	0	0,28465	0,284649461	0,284649	-0,715351
-0,09046	-0,0779465	-0,1397016	0	0,51017	0,510170466	-0,48983	-0,48983

$\begin{bmatrix} \Delta\mathbf{q}^{(k)} \\ \Delta\mathbf{H}^{(k)} \end{bmatrix}$

			0	0,0217645	0,021764
$\Delta\mathbf{M}^{(k)}$	0		8,364E-14	-1,815E-06	-1,8E-06
	-1E-13		8,364E-14	-1,815E-06	-1,8E-06
	2,3E-13		-1,437E-13	-1,815E-06	-1,8E-06
	0,01339		-3,005E-14	-1,815E-06	-1,8E-06
	-153,45		-1,483E-14	-153,45415	-153,454
$\Delta\mathbf{h}^{(k)}$	-40,83		-4,782E-15	-121,24352	-121,244
	73,041	12,04924753	-2,29E-14	-194,28448	-194,284
	121,244				

	jacobiano(nxm)									
iteration:6	index	H| q	1	2	3	4	5	2	3	4
	2	1874,8838	0	1	-1	0	0	0	0	0
	3	1374,5851	0	0	0	1	-1	0	0	0
	4	333,41078	0	0	1	-1	0	0	0	0
	1	-6917,147	0,615138	0	0	0	0	0	0	0
	2	1600,9837	0	0,1773223	0	0	0	1	0	0
	3	782,66945	0	0	0,0965109	0	0	-1	0	1
	4	-1071,791	0	0	0	0,126075	0	0	1	-1
	5	-1409,544	0	0	0	0	0,15913	0	-1	0

Inverse Jacobian							
0	0	0	1,625652	0	0	0	0
0,682808	0,2846495	0,5101705	0	1,78879	1,788787947	1,788788	1,7887879
-0,31719	0,2846495	0,5101705	0	1,78879	1,788787947	1,788788	1,7887879
-0,31719	0,2846495	-0,4898295	0	1,78879	1,788787947	1,788788	1,7887879
-0,31719	-0,7153505	-0,4898295	0	1,78879	1,788787947	1,788788	1,7887879
-0,12108	-0,0504747	-0,0904646	0	0,68281	-0,31719206	-0,317192	-0,317192
-0,05047	-0,1138336	-0,0779465	0	0,28465	0,284649461	0,284649	-0,715351
-0,09046	-0,0779465	-0,1397016	0	0,51017	0,510170466	-0,48983	-0,48983

$\begin{bmatrix} \Delta\mathbf{q}^{(k)} \\ \Delta\mathbf{H}^{(k)} \end{bmatrix}$

			0	2,911E-08	2,91E-08
$\Delta\mathbf{M}^{(k)}$	-1E-13		-1,1E-13	-3,979E-13	-5,1E-13
	-1E-13		3,7E-15	-3,979E-13	-3,9E-13
	0		3,7E-15	-3,979E-13	-3,9E-13
	1,8E-08		1,174E-13	-3,979E-13	-2,8E-13
	-153,45		1,95E-14	-153,45415	-153,454
$\Delta\mathbf{h}^{(k)}$	-40,83		1,868E-14	-121,24352	-121,244
	73,041	1,34E-02	1,915E-14	-194,28448	-194,284
	121,244				

Iteration 7 and the optimization process:

	jacobiano(nxm)									
iteration:7	index	H| q	1	2	3	4	5	2	3	4
	2	1721,43	0	1	-1	0	0	0	0	0
	3	1253,342	0	0	0	1	-1	0	0	0
	4	139,1263	0	0	1	-1	0	0	0	0
	1	-6917,15	0,615138	0	0	0	0	0	0	0
	2	1600,984	0	0,1773223	0	0	0	1	0	0
	3	782,6695	0	0	0,096511	0	0	-1	0	1
	4	-1071,79	0	0	0	0,12607	0	0	1	-1
	5	-1409,54	0	0	0	0	0,159129796	0	-1	0

Inverse Jacobian							
0	0	0	1,62565	0	0	0	0
0,6828079	0,2846495	0,51017	0	1,788787947	1,788788	1,7887879	1,788787947
-0,3171921	0,2846495	0,51017	0	1,788787947	1,788788	1,7887879	1,788787947
-0,3171921	0,2846495	-0,48983	0	1,788787947	1,788788	1,7887879	1,788787947
-0,3171921	-0,7153505	-0,48983	0	1,788787947	1,788788	1,7887879	1,788787947
-0,1210771	-0,0504747	-0,09046	0	0,682807936	-0,317192	-0,317192	-0,317192064
-0,0504747	-0,1138336	-0,07795	0	0,284649461	0,284649	0,2846495	-0,715350539
-0,0904646	-0,0779465	-0,1397	0	0,510170466	0,51017	-0,48983	-0,489829534

$$\begin{bmatrix} \Delta \mathbf{q}^{(k)} \\ \Delta \mathbf{H}^{(k)} \end{bmatrix}$$

			0	0	0
$\Delta \mathbf{M}^{(k)}$	0		-8,364E-14	1,99E-13	1,2E-13
	1,137E-13		-8,364E-14	1,99E-13	1,2E-13
	-2,27E-13		1,437E-13	1,99E-13	3,4E-13
	0		3,005E-14	1,99E-13	2,3E-13
	-153,4542		1,483E-14	-153,454	-153,45
$\Delta \mathbf{h}^{(k)}$	-40,83033		4,782E-15	-121,244	-121,24
	73,040962	1,79E-08	2,29E-14	-194,284	-194,28
	121,24352				

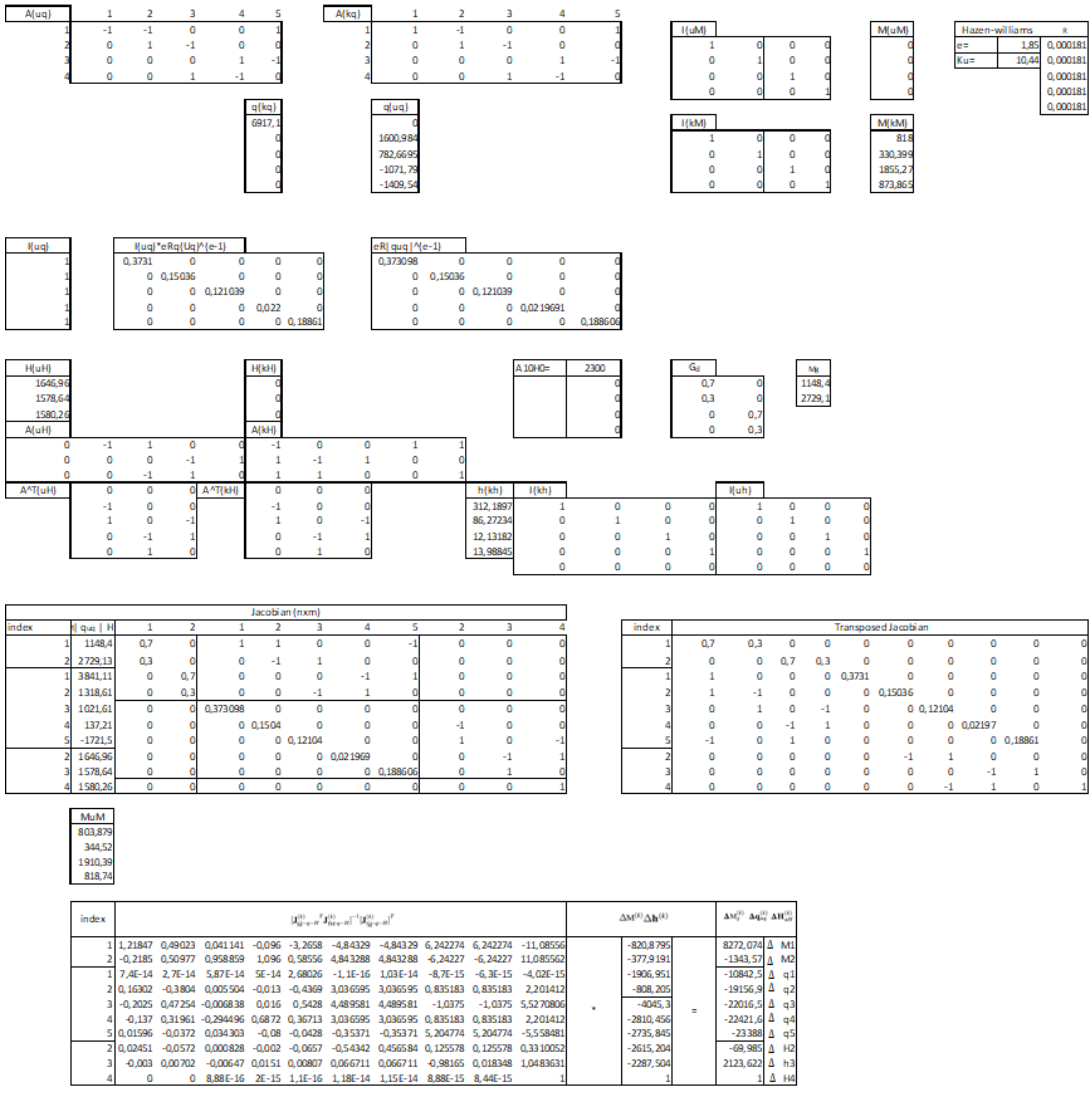
Observado exp.		Optimização	
index	H|	obj=erro=desvio	
2	1721,432		5,765E-06
3	1253,342		1,433E-07
4	139,1243		3,918E-06

(nodes)	M: Demand
1	818,3142631
2	337,7534207
3	1854,460374
4	810,6042936

erro final	9,826E-06

Model Inverse method:

$$\mathbf{Aq} - \mathbf{M} = 0 \quad \text{and} \quad [\mathbf{I}_{kM} \quad \mathbf{I}_{uM}] \cdot \begin{bmatrix} \mathbf{M}_{kM} \\ \mathbf{M}_{uM} \end{bmatrix} - [\mathbf{A}_{kq} \quad \mathbf{A}_{uq}] \cdot \begin{bmatrix} \mathbf{q}_{kq} \\ \mathbf{q}_{uq} \end{bmatrix} = 0$$



A(uq)	1	2	3	4	5
1	-1	-1	0	0	1
2	0	1	-1	0	0
3	0	0	0	1	-1
4	0	0	1	-1	0

A(kq)	1	2	3	4	5
1	1	-1	0	0	1
2	0	1	-1	0	0
3	0	0	0	1	-1
4	0	0	1	-1	0

I(uM)			
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

M(uM)
0
0
0
0

Hazen-williams	x
e=	1,85
Ku=	10,44

	x
	0,000181
	0,000181
	0,000181
	0,000181
	0,000181

q(kq)
6917,1
0
0
0
0

q(uq)
0
1600,984
782,6695
-1071,79
-1409,54

I(kM)			
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

M(kM)
818
330,399
1855,27
873,865

I(uq)
1
1
1
1
1

I(uq)*eRq(Uq)^(e-1)				
0,3731	0	0	0	0
0	0,15036	0	0	0
0	0	0,121039	0	0
0	0	0	0,022	0
0	0	0	0	0,18861

eR(quq)^(e-1)				
0,373098	0	0	0	0
0	0,15036	0	0	0
0	0	0,121039	0	0
0	0	0	0,021969	0
0	0	0	0	0,188606

H(uH)
1646,96
1578,64
1580,26

H(kH)
0
0
0

A 10H0=	2300
	0
	0
	0
	0

G_{ij}	
0,7	0
0,3	0
0	0,7
0	0,3

M_K
1148,4
2729,1

A(uH)									
0	-1	1	0	0	-1	0	0	1	1
0	0	0	-1	1	1	-1	1	0	0
0	0	-1	1	0	1	1	0	0	1

A(kH)									
0	-1	1	0	0	-1	0	0	1	1
0	0	0	-1	1	1	-1	1	0	0
0	0	-1	1	0	1	1	0	0	1

A^T(uH)			
0	0	0	
-1	0	0	
1	0	-1	
0	-1	1	
0	1	0	

A^T(kH)			
0	0	0	
-1	0	0	
1	0	-1	
0	-1	1	
0	1	0	

h(kh)
312,1897
86,27234
12,13182
13,98845

I(kh)				
1	0	0	0	
0	1	0	0	
0	0	1	0	
0	0	0	1	
0	0	0	0	

I(uh)				
1	0	0	0	
0	1	0	0	
0	0	1	0	
0	0	0	1	
0	0	0	0	

Jacobian (nxm)

index	I(q_{uq}) I H	1	2	1	2	3	4	5	2	3	4
1	1148,4	0,7	0	1	1	0	0	-1	0	0	0
2	2729,13	0,3	0	0	-1	1	0	0	0	0	0
1	3841,11	0	0,7	0	0	0	-1	1	0	0	0
2	1318,61	0	0,3	0	0	-1	1	0	0	0	0
3	1021,61	0	0	0,373098	0	0	0	0	0	0	0
4	137,21	0	0	0	0,15104	0	0	0	-1	0	0
5	-1721,5	0	0	0	0	0,12104	0	0	1	0	-1
2	1646,96	0	0	0	0	0	0,021969	0	0	-1	1
3	1578,64	0	0	0	0	0	0	0,188606	0	1	0
4	1580,26	0	0	0	0	0	0	0	0	0	1

Transposed Jacobian

index										
1	0,7	0,3	0	0	0	0	0	0	0	0
2	0	0	0,7	0,3	0	0	0	0	0	0
1	1	0	0	0	0,3731	0	0	0	0	0
2	1	-1	0	0	0	0,15036	0	0	0	0
3	0	1	0	-1	0	0	0,12104	0	0	0
4	0	0	-1	1	0	0	0	0,02197	0	0
5	-1	0	1	0	0	0	0	0	0,18861	0
2	0	0	0	0	0	-1	1	0	0	0
3	0	0	0	0	0	0	0	-1	1	0
4	0	0	0	0	0	0	-1	1	0	1

MuM
803,879
344,52
1910,39
818,74

index														
1	1,21847	0,49023	0,041141	-0,096	-3,2658	-4,84329	-4,84329	6,242274	6,242274	-11,08556		-820,8795		8272,074	Δ	M1
2	-0,2185	0,50977	0,958859	1,096	0,58556	4,843288	4,843288	-6,24227	-6,24227	11,085562		-377,9191		-1343,57	Δ	M2
1	7,4E-14	2,7E-14	5,87E-14	5E-14	2,68026	-1,1E-16	1,03E-14	-8,7E-15	-6,3E-15	-4,02E-15		-1906,951		-10842,5	Δ	q1
2	0,16302	-0,3804	0,005504	-0,013	-0,4369	3,036595	3,036595	0,835183	0,835183	2,201412		-808,205		-19156,9	Δ	q2
3	-0,2025	0,47254	-0,006838	0,016	0,5428	4,489581	4,489581	-1,0375	-1,0375	5,527080	*	-4045,3	=	-22016,5	Δ	q3
4	-0,137	0,31961	-0,294496	0,6872	0,36713	3,036595	3,036595	0,835183	0,835183	2,201412		-2810,456		-22421,6	Δ	q4
5	0,01596	-0,0372	0,034303	-0,08	-0,0428	-0,35371	-0,35371	5,204774	5,204774	-5,558481		-2735,845		-23388	Δ	q5
2	0,02451	-0,0572	0,000828	-0,002	-0,0657	-0,54342	0,456584	0,125578	0,125578	0,331005		-2615,204		-69,985	Δ	H2
3	-0,003	0,00702	-0,00647	0,0151	0,00807	0,066711	0,066711	-0,98165	0,018348	1,048363		-2287,504		2123,622	Δ	h3
4	0	0	8,88E-16	2E-15	1,1E-16	1,18E-14	1,15E-14	8,88E-15	8,44E-15	1		1		1	Δ	H4

Iteration 1:

index	M_g | q_{uq} | H_{uH}
1	9420,474
2	1385,566
1	-7001,35
2	-17838,3
3	-20994,8
4	-22284,4
5	-25109,5
2	1576,975
3	3702,262
4	1581,26

MuM
6594,332
2826,142
969,8959
415,6697

R\|quq\|^(e-a)

					I(uq)	I(uq)*Rq(Uq)				
0,33594	0	0	0	0	1	0,335944	0	0	0	0
0	0,743898	0	0	0	1	0	0,743898	0	0	0
0	0	0,8543949	0	0	1	0	0	0,85439	0	0
0	0	0	0,8988	0	1	0	0	0	0,898801	0
0	0	0	0	0,994777	1	0	0	0	0	0,9947773

0,7	0	1	1	0	0	-1	0	0	0
0,3	0	0	-1	1	0	0	0	0	0
0	0,7	0	0	0	-1	1	0	0	0
0	0,3	0	0	-1	1	0	0	0	0
0	0	0,3359444	0	0	0	0	0	0	0
0	0	0	0,7439	0	0	0	-1	0	0
0	0	0	0	0,854395	0	0	1	0	-1
0	0	0	0	0	0,898801	0	0	-1	1
0	0	0	0	0	0	0,9947773	0	1	0
0	0	0	0	0	0	0	0	0	1

0,7	0,3	0	0	0	0	0	0	0	0
0	0	0,7	0,3	0	0	0	0	0	0
1	0	0	0	0,335944	0	0	0	0	0
1	-1	0	0	0	0,743898	0	0	0	0
0	1	0	-1	0	0	0,8543949	0	0	0
0	0	-1	1	0	0	0	0,898801	0	0
-1	0	1	0	0	0	0	0	0,994777	0
0	0	0	0	0	-1	1	0	0	0
0	0	0	0	0	0	0	-1	1	0
0	0	0	0	0	0	-1	1	0	1

inversa (jacobiano transp*jacobiano)

1,2644	0,383077	0,2696355	-0,6291	-3,7637	-1,18473	-1,18473	0,999983	0,999983	-2,1847
-0,2644	0,616923	0,7303645	1,62915	0,787021	1,18473	1,1847295	-0,99998	-0,999983	2,18471
3E-15	3,11E-15	8,66E-15	1,6E-15	2,976683	1,55E-15	1,924E-15	1,96E-15	-2,89E-15	5,2E-16
0,20277	-0,47313	0,0432415	-0,1009	-0,60359	0,435672	0,4356723	0,160367	0,160367	0,2753
-0,17655	0,411944	-0,037649	0,08785	0,525525	0,791091	0,7910912	-0,13963	-0,139627	0,93072
-0,09723	0,226867	-0,256759	0,5991	0,289419	0,435672	0,4356723	0,160367	0,160367	0,2753
0,08785	-0,20498	0,2319863	-0,5413	-0,2615	-0,39364	-0,393638	0,860355	0,860355	-1,254
0,15084	-0,35196	0,0321672	-0,0751	-0,44901	-0,6759	0,3240958	0,119297	0,119297	0,2048
-0,08739	0,203908	-0,230775	0,53847	0,26013	0,391583	0,3915825	-0,85586	0,144138	1,24744
-2,2E-16	3,33E-16	-2,22E-16	5,6E-16	7,74E-16	9,16E-16	4,441E-16	-4,4E-16	-8,88E-16	1

-3841,11	delta Mg	4269,59
3,8E-11		-7336,4
-3E-11		-774,31
6,1E-11		19820,4
-260,124		18539,5
12957,7	delta q_{uq}	20740,4
17625,7		25875,9
19717		1786,65
24666,2		-1074,5
1	delta H_{uH}	1

Iterations 2 - 4:

iter2

index	Mg | quq | Hun
1	13690,06
2	-5950,822
1	-7775,66
2	1982,069
3	-2455,348
4	-1543,966
5	766,3409
2	3363,622
3	2627,733
4	1582,26

MuM
9583,041
4107,018
-4165,575
-1785,247

R|quq|^(e-a)					I(uq)	I(uq)*Rq(Uq)				
0,367273	0	0	0	0	1	0,36727	0	0	0	0
0	0,114925	0	0	0	1	0	0,114925	0	0	0
0	0	0,137867	0	0	1	0	0	0,138	0	0
0	0	0	0,0929405	0	1	0	0	0	0,0929	0
0	0	0	0	0,0512415	1	0	0	0	0	0,05124

0,7	0	1	1	0	0	-1	0	0	0
0,3	0	0	-1	1	0	0	0	0	0
0	0,7	0	0	0	-1	1	0	0	0
0	0,3	0	0	-1	1	0	0	0	0
0	0	0,367273	0	0	0	0	0	0	0
0	0	0	0,1149248	0	0	0	-1	0	0
0	0	0	0	0,1378666	0	0	1	0	-1
0	0	0	0	0	0,0929405	0	0	-1	1
0	0	0	0	0	0	0,0512415	0	1	0
0	0	0	0	0	0	0	0	0	1

iter3

index	Mg | quq | Hun
1	10022,62
2	-2946,654
1	-7112,384
2	-1084,083
3	-4421,267
4	-4411,136
5	-4203,746
2	3323,448
3	2530,95
4	1583,26

MuM
7015,831
3006,785
-2062,658
-883,9962

R|quq|^(e-a)					I(uq)	I(uq)*Rq(Uq)				
0,340468	0	0	0	0	1	0,34047	0	0	0	0
0	0,068812	0	0	0	1	0	0,068812	0	0	0
0	0	0,227289	0	0	1	0	0	0,227	0	0
0	0	0	0,2268458	0	1	0	0	0	0,2268	0
0	0	0	0	0,2177478	1	0	0	0	0	0,21775

0,7	0	1	1	0	0	-1	0	0	0
0,3	0	0	-1	1	0	0	0	0	0
0	0,7	0	0	0	-1	1	0	0	0
0	0,3	0	0	-1	1	0	0	0	0
0	0	0,340468	0	0	0	0	0	0	0
0	0	0	0,0688123	0	0	0	-1	0	0
0	0	0	0	0,2272886	0	0	1	0	-1
0	0	0	0	0	0,2268458	0	0	-1	1
0	0	0	0	0	0	0,2177478	0	1	0
0	0	0	0	0	0	0	0	0	1

iter4

index	Mg | quq | Hun
1	1150,235
2	2699,284
1	7672,359
2	1148,444
3	-3090,373
4	-2346,602
5	-427,3867
2	3714,664
3	2311,823
4	1584,26

R|quq|^(e-a)					I(uq)	I(uq)*Rq(Uq)				
0,363122	0	0	0	0	1	0,36312	0	0	0	0
0	0,07227	0	0	0	1	0	0,07227	0	0	0
0	0	0,167638	0	0	1	0	0	0,168	0	0
0	0	0	0,1326589	0	1	0	0	0	0,1327	0
0	0	0	0	0,0311933	1	0	0	0	0	0,03119

0,7	0	1	1	0	0	-1	0	0	0
0,3	0	0	-1	1	0	0	0	0	0
0	0,7	0	0	0	-1	1	0	0	0
0	0,3	0	0	-1	1	0	0	0	0
0	0	0,363122	0	0	0	0	0	0	0
0	0	0	0,0722697	0	0	0	-1	0	0
0	0	0	0	0,1676378	0	0	1	0	-1
0	0	0	0	0	0,1326589	0	0	-1	1
0	0	0	0	0	0	0,0311933	0	1	0
0	0	0	0	0	0	0	0	0	1

Gd	
0,7	0
0,3	0
0	0,7
0	0,3

MuM
805,1644
345,0704
1889,499
809,7853

iterations	Node 1	Node 2	Node 3	Node 4
0	803,88	344,52	1910,39	818,74
1	6594,33	2826,14	969,90	415,67
2	9583,04	4107,02	-4165,58	-1785,25
3	7015,83	3006,78	-2062,66	-884,00
4	805,16	345,07	1889,50	809,79

Iterations 2 – 4 (continuation):

iter2

0,7	0,3	0	0	0	0	0	0	0	0
0	0	0,7	0,3	0	0	0	0	0	0
1	0	0	0	0,367273	0	0	0	0	0
1	-1	0	0	0	0,1149248	0	0	0	0
0	1	0	-1	0	0	0,13787	0	0	0
0	0	-1	1	0	0	0	0,09294	0	0
-1	0	1	0	0	0	0	0	0,051241	0
0	0	0	0	0	-1	1	0	0	0
0	0	0	0	0	0	0	-1	1	0
0	0	0	0	0	0	-1	1	0	1

$\cdot \begin{bmatrix} \Delta M^{(k)} \\ \Delta \mathbf{h}^{(k)} \end{bmatrix}$

inversa (jacobiano transp*jacobiano)										iter2			
1,330724	0,228311	0,46893	-1,09417	-3,62325	-9,592472	-9,5925	16,81829	16,81829	-26,41076	1,09E-11		delta Mg	-3667,444
-0,33072	0,771689	0,53107	2,09417	0,900484	9,5924719	9,59247	-16,8183	-16,8183	26,410762	2,18E-11	1,1E-10		3004,168
5,66E-14	2,13E-13	1,58E-13	-9,3E-14	2,722767	1,407E-14	2,9E-14	3,07E-15	-1,5E-14	1,846E-14	3,46E-11			663,27567
0,217724	-0,508022	0,076723	-0,17902	-0,59281	2,3863777	2,38638	2,751692	2,751692	-0,365314	1,57E-11			-3066,152
-0,18149	0,423485	-0,06396	0,14923	0,494164	5,2641192	5,26412	-2,2938	-2,2938	7,5579147	243,6035			-1965,919
-0,08228	0,191978	-0,22328	0,52098	0,224019	2,3863777	2,38638	2,751692	2,751692	-0,365314	-312,2029		delta q_{uq}	-2867,17
0,149231	-0,348205	0,404974	-0,94494	-0,40632	-4,328353	-4,3284	14,52449	14,52449	-18,85285	-312,2087			-4970,087
0,025022	-0,058384	0,008817	-0,02057	-0,06813	-0,725746	0,27425	0,316238	0,316238	-0,041984	-168,6926			-40,17415
-0,00765	0,017843	-0,02075	0,04842	0,02082	0,2217911	0,22179	-0,74426	0,255744	0,9660475	-351,4581			-96,78353
-3,6E-15	1,78E-15	3,55E-15	3,2E-15	6,87E-15	2,771E-14	2,8E-14	-2,7E-14	-2,7E-14	1	1		delta H_{uH}	1

iter3

0,7	0,3	0	0	0	0	0	0	0	0
0	0	0,7	0,3	0	0	0	0	0	0
1	0	0	0	0,340468	0	0	0	0	0
1	-1	0	0	0	0,0688123	0	0	0	0
0	1	0	-1	0	0	0,22729	0	0	0
0	0	-1	1	0	0	0	0,226846	0	0
-1	0	1	0	0	0	0	0	0,217748	0
0	0	0	0	0	-1	1	0	0	0
0	0	0	0	0	0	0	-1	1	0
0	0	0	0	0	0	-1	1	0	1

$\cdot \begin{bmatrix} \Delta M^{(k)} \\ \Delta \mathbf{h}^{(k)} \end{bmatrix}$

inversa (jacobiano transp*jacobiano)										iter3			
1,121647	0,716156	0,267081	-0,62319	-3,29443	-5,892713	-5,8927	3,924567	3,924567	-9,817279	9,09E-13		delta Mg	3005,4405
-0,12165	0,283844	0,732919	1,62319	0,357295	5,8927127	5,89271	-3,92457	-3,92457	9,8172792	4,55E-13	4,8E-11		-2445,465
2,44E-14	-9,66E-15	-1,1E-13	-8,3E-15	2,937137	-5,38E-15	-5E-15	1,53E-16	-1,8E-14	-2,33E-15	9,09E-13			-559,9754
0,258295	-0,602688	0,061504	-0,14351	-0,75865	2,0202445	2,02024	0,903755	0,903755	1,1164892	1,14E-12			2232,5266
-0,0782	0,182465	-0,01862	0,04345	0,229683	3,7880583	3,78806	-0,27361	-0,27361	4,061673	-190,6535			1330,8945
-0,04171	0,097312	-0,2385	0,55649	0,122494	2,0202445	2,02024	0,903755	0,903755	1,1164892	-237,5914		delta q_{uq}	2064,534
0,043448	-0,101378	0,248461	-0,57974	-0,12761	-2,104654	-2,1047	3,650952	3,650952	-5,755606	692,7138			3776,3596
0,017774	-0,041472	0,004232	-0,00988	-0,0522	-0,860982	0,13902	0,062189	0,062189	0,0768282	688,458			391,21669
-0,00946	0,022075	-0,0541	0,12624	0,027787	0,458284	0,45828	-0,79499	0,205013	1,2532709	603,1671			-219,1271
8,88E-16	-8,88E-16	0	1,1E-15	-1E-15	7,633E-16	2,2E-16	-2,2E-15	-2E-15	1	1		delta H_{uH}	1

iter4

0,7	0,3	0	0	0	0	0	0	0	0
0	0	0,7	0,3	0	0	0	0	0	0
1	0	0	0	0,363122	0	0	0	0	0
1	-1	0	0	0	0,0722697	0	0	0	0
0	1	0	-1	0	0	0,16764	0	0	0
0	0	-1	1	0	0	0	0,132659	0	0
-1	0	1	0	0	0	0	0	0,031193	0
0	0	0	0	0	-1	1	0	0	0
0	0	0	0	0	0	0	-1	1	0
0	0	0	0	0	0	-1	1	0	1

$\cdot \begin{bmatrix} \Delta M^{(k)} \\ \Delta \mathbf{h}^{(k)} \end{bmatrix}$

inversa (jacobiano transp*jacobiano)										iter4			
1,263559	0,385028	0,708354	-1,65283	-3,47971	-12,15629	-12,156	17,79889	17,79889	-29,95518	-7030,244		delta Mg	8209,1826
-0,26356	0,614972	0,291646	2,65283	0,725815	12,156292	12,1563	-17,7989	-17,7989	29,955178	3563,346	6E-12		-4901,425
-1,6E-13	4,52E-14	-2,7E-13	-2,9E-13	2,753896	1,196E-14	1,8E-14	1,39E-14	-7,2E-15	2,228E-14	-5663,982			-14866,06
0,264878	-0,618048	0,148491	-0,34648	-0,72945	1,6199728	1,61997	3,731147	3,731147	-2,111174	-2427,421			-1553,992
-0,11419	0,266444	-0,06402	0,14937	0,314468	5,2668604	5,26686	-1,60852	-1,60852	6,8753798	-5398,191			-453,4004
-0,03512	0,081952	-0,15151	0,35352	0,096724	1,6199728	1,61997	3,731147	3,731147	-2,111174	-395,1873		delta q_{uq}	-1410,394
0,149369	-0,348528	0,644339	-1,50346	-0,41135	-6,889432	-6,8894	16,19037	16,19037	-23,0798	205,8737			-3643,379
0,019143	-0,044666	0,010731	-0,02504	-0,05272	-0,882925	0,11707	0,269649	0,269649	-0,152574	-0,892045			282,8808
-0,00466	0,010872	-0,0201	0,0469	0,012831	0,2149038	0,2149	-0,50503	0,49497	0,719934	-298,8581			-185,2092
3,55E-15	-2,66E-15	3,55E-15	0	-1,1E-15	4,913E-15	5E-15	3,22E-14	3,23E-14	1	1		delta H_{uH}	1

B. Transient Operations and Results

Table 37. Tank's water level: Inverse model and Epanet comparison

Hour	Inverse model	Epanet
0	2529.00	2529.00
1	9604.38	8992.00
2	24124.99	24166.00
3	12572.02	15736.00
4	19904.62	19670.00
5	17838.16	16860.00
6	25067.50	25290.00
7	19800.72	19108.00
8	23003.32	23604.00
9	24242.82	24728.00
10	27134.04	25571.00
11	24965.65	24447.00
12	19180.75	19389.00
13	19697.32	19501.40
14	17114.66	17422.00
15	24903.51	24278.40
16	25110.26	24166.00
17	19738.63	19951.00
18	20771.78	20007.20
19	21805.12	21440.30
20	25929.81	26048.70
21	38275.23	39902.00
22	24583.06	25009.00
23	26038.94	26133.00

Table 38. Flow rate (q_{in}) for the Transient case

Hour	q_{in} (t)
0	18000
1	690000
2	2099000
3	980000
4	1690000
5	1490000
6	2190000
7	1680000
8	1990000
9	2110000
10	2390000
11	2180000
12	1620000
13	1670000
14	1420000
15	2174000
16	2194000
17	1674000
18	1774000
19	1874000
20	2274000
21	3474000
22	2143000
23	2284000

Table 39. Pattern consumption multipliers for the Transient Case

Hour	Node 1	Node 2	Node 3	Node 4
0	0.0639	0.0639	0.53106	0.12
1	0.2272	0.2272	0.2014	0.18
2	0.6106	0.6106	0.7632	0.71
3	0.3976	0.3976	0.02438	0.01
4	0.497	0.497	0.1113	0.103
5	0.426	0.426	0.2544	0.22
6	0.639	0.639	0.2809	0.26
7	0.4828	0.4828	0.1961	0.192
8	0.5964	0.5964	0.053	0.047
9	0.6248	0.6248	0.1272	0.121
10	0.6461	0.6461	0.1537	0.16
11	0.6177	0.6177	0.0742	0.066
12	0.4899	0.4899	0.2226	0.22
13	0.49274	0.49274	0.212	0.2
14	0.4402	0.4402	0.3074	0.28
15	0.61344	0.61344	0.1007	0.093
16	0.6106	0.6106	0.0742	0.07
17	0.5041	0.5041	0.1908	0.2
18	0.50552	0.50552	0.1696	0.164
19	0.54173	0.54173	0.106	0.082
20	0.65817	0.65817	0.2438	0.22
21	1.0082	1.0082	1.007	1
22	0.6319	0.6319	0.1219	0.12
23	0.6603	0.6603	0.1908	0.18

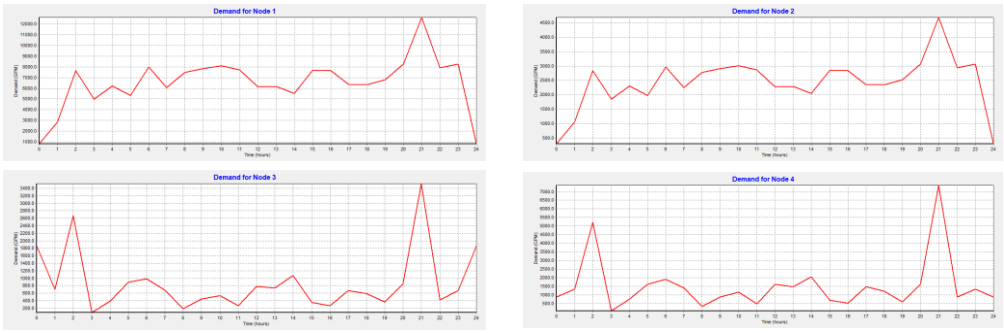


Fig. 15 Nodal demands graphs for the Transient case (EPANET)

Inverse model application within the first six hours: [00:00 h]

$$\mathbf{Aq} - \mathbf{M} = 0 \quad \text{and} \quad [\mathbf{I}_{kM} \quad \mathbf{I}_{uM}] \cdot \begin{bmatrix} \mathbf{M}_{kM} \\ \mathbf{M}_{uM} \end{bmatrix} - [\mathbf{A}_{kq} \quad \mathbf{A}_{uq}] \cdot \begin{bmatrix} \mathbf{q}_{kq} \\ \mathbf{q}_{uq} \end{bmatrix} = 0$$

A(uq)	1	2	3	4	5
1	-1	-1	0	0	1
2	0	1	-1	0	0
3	0	0	0	1	-1
4	0	0	1	-1	0

A(kq)	1	2	3	4	5
1	1	-1	0	0	1
2	0	1	-1	0	0
3	0	0	0	1	-1
4	0	0	1	-1	0

I(uM)			
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

M(uM)
0
0
0
0

Hazen-williams		R
e=	1,85	0,000181
Ku=	10,44	0,000181
		0,000181
		0,000181
		0,000181

q(kq)
6917,1
0
0
0
0

q(uq)
0
1600,984
782,6695
-1071,79
-1409,54

I(kM)			
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

M(kM)
818
330,399
1855,27
873,865

I(uq)
1
1
1
1
1

I(uq)*eRq(Uq)^(e-1)				
0,3731	0	0	0	0
0	0,15036	0	0	0
0	0	0,121039	0	0
0	0	0	0,022	0
0	0	0	0	0,18861

eR|quq|^(e-1)				
0,373098	0	0	0	0
0	0,15036	0	0	0
0	0	0,1210394	0	0
0	0	0	0,0219691	0
0	0	0	0	0,188606

H(uH)
1646,96
1578,64
1580,26

H(kH)
0
0
0

A10H0=	2300
	0
	0
	0
	0

Gd	
0,7	0
0,3	0
0	0,7
0	0,3

Mg
1148,4
2729,1

A(uH)					A(kH)					
0	-1	1	0	0	-1	0	0	1	1	
0	0	0	-1	1	1	-1	1	0	0	
0	0	-1	1	0	1	1	0	0	1	

A^T(uH)			
0	0	0	
-1	0	0	
1	0	-1	
0	-1	1	
0	1	0	

A^T(kH)			
0	0	0	
-1	0	0	
1	0	-1	
0	-1	1	
0	1	0	

h(kh)	I(kh)				I(uh)			
312,18965	1	0	0	0	1	0	0	0
86,27234	0	1	0	0	0	1	0	0
12,131818	0	0	1	0	0	0	1	0
13,98845	0	0	0	1	0	0	0	1
	0	0	0	0	0	0	0	0

Jacobian (nxm)											
index	| q_{uq} | H	1	2	1	2	3	4	5	2	3	4
1	1148,4	0,7	0	1	1	0	0	-1	0	0	0
2	2729,13	0,3	0	0	-1	1	0	0	0	0	0
1	3841,11	0	0,7	0	0	0	-1	1	0	0	0
2	1318,61	0	0,3	0	0	-1	1	0	0	0	0
3	1021,61	0	0	0,373098	0	0	0	0	0	0	0
4	137,21	0	0	0	0,1504	0	0	0	-1	0	0
5	-1721,5	0	0	0	0	0,12104	0	0	1	0	-1
2	1646,96	0	0	0	0	0	0,021969	0	0	-1	1
3	1578,64	0	0	0	0	0	0	0,188606	0	1	0
4	1580,26	0	0	0	0	0	0	0	0	0	1

index	Transposed Jacobian									
1	0,7	0,3	0	0	0	0	0	0	0	0
2	0	0	0,7	0,3	0	0	0	0	0	0
1	1	0	0	0	0,3731	0	0	0	0	0
2	1	-1	0	0	0	0,15036	0	0	0	0
3	0	1	0	-1	0	0	0,12104	0	0	0
4	0	0	-1	1	0	0	0	0,02197	0	0
5	-1	0	1	0	0	0	0	0	0,18861	0
2	0	0	0	0	0	-1	1	0	0	0
3	0	0	0	0	0	0	0	-1	1	0
4	0	0	0	0	0	0	-1	1	0	1

MuM
803,879
344,52
1910,39
818,74

index	$[\mathbf{J}^{(k)}_{M^{k-H}}]^T \mathbf{s}^{(k)}_{M^k-H}]^{-1} [\mathbf{J}^{(k)}_{M^k-H}]^T$									$\Delta \mathbf{M}^{(k)} \Delta \mathbf{h}^{(k)}$		$\Delta \mathbf{M}^{(k)}_{kj}$ $\Delta \mathbf{q}^{(k)}_{uq}$ $\Delta \mathbf{H}^{(k)}_{uH}$	
1	1,21847	0,49023	0,041141	-0,096	-3,2658	-4,84329	-4,84329	6,242274	6,2422741	-11,08556	-820,8795	8272,074	Δ M1
2	-0,2185	0,50977	0,958859	1,096	0,58556	4,843288	4,843288	-6,24227	-6,242274	11,085562	-377,9191	-1343,57	Δ M2
1	7,4E-14	2,7E-14	5,87E-14	5E-14	2,68026	-1,1E-16	1,03E-14	-8,7E-15	-6,27E-15	-4,02E-15	-1906,951	-10842,5	Δ q1
2	0,16302	-0,3804	0,005504	-0,013	-0,4369	3,036595	3,036595	0,835183	0,8351826	2,201412	-808,205	-19156,9	Δ q2
3	-0,2025	0,47254	-0,006838	0,016	0,5428	4,489581	4,489581	-1,0375	-1,0375	5,5270806	-4045,3	-22016,5	Δ q3
4	-0,137	0,31961	-0,294496	0,6872	0,36713	3,036595	3,036595	0,835183	0,8351826	2,201412	-2810,456	-22421,6	Δ q4
5	0,01596	-0,0372	0,034303	-0,08	-0,0428	-0,35371	-0,35371	5,204774	5,2047744	-5,558481	-2735,845	-23388	Δ q5
2	0,02451	-0,0572	0,000828	-0,002	-0,0657	-0,54342	0,456584	0,125578	0,1255784	0,3310052	-2615,204	-69,985	Δ H2
3	-0,003	0,00702	-0,00647	0,0151	0,00807	0,066711	0,066711	-0,98165	0,0183482	1,0483631	-2287,504	2123,622	Δ h3
4	0	0	8,88E-16	2E-15	1,1E-16	1,18E-14	1,15E-14	8,88E-15	8,438E-15	1	1	1	Δ H4

index	M_g | q_{uq} | H_{uH}
1	9420,474
2	1385,566
1	-7001,35
2	-17838,3
3	-20994,8
4	-22284,4
5	-25109,5
2	1576,975
3	3702,262
4	1581,26

MuM
6594,332
2826,142
969,8959
415,6697

R|quq|^(e-a)					I(uq)	I(uq)*Rq(Uq)				
0,33594	0	0	0	0	1	0,335944	0	0	0	0
0	0,743898	0	0	0	1	0	0,743898	0	0	0
0	0	0,8543949	0	0	1	0	0	0,85439	0	0
0	0	0	0,8988	0	1	0	0	0	0,898801	0
0	0	0	0	0,994777	1	0	0	0	0	0,9947773

0,7	0	1	1	0	0	-1	0	0	0
0,3	0	0	-1	1	0	0	0	0	0
0	0,7	0	0	0	-1	1	0	0	0
0	0,3	0	0	-1	1	0	0	0	0
0	0	0,3359444	0	0	0	0	0	0	0
0	0	0	0,7439	0	0	0	-1	0	0
0	0	0	0	0,854395	0	0	1	0	-1
0	0	0	0	0	0,898801	0	0	-1	1
0	0	0	0	0	0	0,9947773	0	1	0
0	0	0	0	0	0	0	0	0	1

0,7	0,3	0	0	0	0	0	0	0	0
0	0	0,7	0,3	0	0	0	0	0	0
1	0	0	0	0,335944	0	0	0	0	0
1	-1	0	0	0	0,743898	0	0	0	0
0	1	0	-1	0	0	0,8543949	0	0	0
0	0	-1	1	0	0	0	0,898801	0	0
-1	0	1	0	0	0	0	0	0,994777	0
0	0	0	0	0	-1	1	0	0	0
0	0	0	0	0	0	0	-1	1	0
0	0	0	0	0	0	-1	1	0	1

inversa (jacobiano transp*jacobiano)									
1,2644	0,383077	0,2696355	-0,6291	-3,7637	-1,18473	-1,18473	0,999983	0,999983	-2,1847
-0,2644	0,616923	0,7303645	1,62915	0,787021	1,18473	1,1847295	-0,99998	-0,999983	2,18471
3E-15	3,11E-15	8,66E-15	1,6E-15	2,976683	1,55E-15	1,924E-15	1,96E-15	-2,89E-15	5,2E-16
0,20277	-0,47313	0,0432415	-0,1009	-0,60359	0,435672	0,4356723	0,160367	0,160367	0,2753
-0,17655	0,411944	-0,037649	0,08785	0,525525	0,791091	0,7910912	-0,13963	-0,139627	0,93072
-0,09723	0,226867	-0,256759	0,5991	0,289419	0,435672	0,4356723	0,160367	0,160367	0,2753
0,08785	-0,20498	0,2319863	-0,5413	-0,2615	-0,39364	-0,393638	0,860355	0,860355	-1,254
0,15084	-0,35196	0,0321672	-0,0751	-0,44901	-0,6759	0,3240958	0,119297	0,119297	0,2048
-0,08739	0,203908	-0,230775	0,53847	0,26013	0,391583	0,3915825	-0,85586	0,144138	1,24744
-2,2E-16	3,33E-16	-2,22E-16	5,6E-16	7,74E-16	9,16E-16	4,441E-16	-4,4E-16	-8,88E-16	1

-3841,11	delta M_g	4269,59
3,8E-11		-7336,4
-3E-11		-774,31
6,1E-11		19820,4
-260,124		18539,5
12957,7	delta q_{uq}	20740,4
17625,7		25875,9
19717		1786,65
24666,2		-1074,5
1	delta H_{uH}	1

iter2

index	Mg| quq | HuH
1	13690,06
2	-5950,822
1	-7775,66
2	1982,069
3	-2455,348
4	-1543,966
5	766,3409
2	3363,622
3	2627,733
4	1582,26

MuM
9583,041
4107,018
-4165,575
-1785,247

R|quq|^(e-a)					I(uq)	I(uq)*Rq(Uq)				
0,367273	0	0	0	0	1	0,36727	0	0	0	0
0	0,114925	0	0	0	1	0	0,114925	0	0	0
0	0	0,137867	0	0	1	0	0	0,138	0	0
0	0	0	0,0929405	0	1	0	0	0	0,0929	0
0	0	0	0	0,0512415	1	0	0	0	0	0,05124

0,7	0	1	1	0	0	-1	0	0	0
0,3	0	0	-1	1	0	0	0	0	0
0	0,7	0	0	0	-1	1	0	0	0
0	0,3	0	0	-1	1	0	0	0	0
0	0	0,367273	0	0	0	0	0	0	0
0	0	0	0,1149248	0	0	0	-1	0	0
0	0	0	0	0,1378666	0	0	1	0	-1
0	0	0	0	0	0,0929405	0	0	-1	1
0	0	0	0	0	0	0,0512415	0	1	0
0	0	0	0	0	0	0	0	0	1

iter3

index	Mg| quq | HuH
1	10022,62
2	-2946,654
1	-7112,384
2	-1084,083
3	-4421,267
4	-4411,136
5	-4203,746
2	3323,448
3	2530,95
4	1583,26

MuM
7015,831
3006,785
-2062,658
-883,9962

R|quq|^(e-a)					I(uq)	I(uq)*Rq(Uq)				
0,340468	0	0	0	0	1	0,34047	0	0	0	0
0	0,068812	0	0	0	1	0	0,068812	0	0	0
0	0	0,227289	0	0	1	0	0	0,227	0	0
0	0	0	0,2268458	0	1	0	0	0	0,2268	0
0	0	0	0	0,2177478	1	0	0	0	0	0,21775

0,7	0	1	1	0	0	-1	0	0	0
0,3	0	0	-1	1	0	0	0	0	0
0	0,7	0	0	0	-1	1	0	0	0
0	0,3	0	0	-1	1	0	0	0	0
0	0	0,340468	0	0	0	0	0	0	0
0	0	0	0,0688123	0	0	0	-1	0	0
0	0	0	0	0,2272886	0	0	1	0	-1
0	0	0	0	0	0,2268458	0	0	-1	1
0	0	0	0	0	0	0,2177478	0	1	0
0	0	0	0	0	0	0	0	0	1

iter4

index	Mg| quq | HuH
1	1150,235
2	2699,284
1	7672,359
2	1148,444
3	-3090,373
4	-2346,602
5	-427,3867
2	3714,664
3	2311,823
4	1584,26

R|quq|^(e-a)					I(uq)	I(uq)*Rq(Uq)				
0,363122	0	0	0	0	1	0,36312	0	0	0	0
0	0,07227	0	0	0	1	0	0,07227	0	0	0
0	0	0,167638	0	0	1	0	0	0,168	0	0
0	0	0	0,1326589	0	1	0	0	0	0,1327	0
0	0	0	0	0,0311933	1	0	0	0	0	0,03119

0,7	0	1	1	0	0	-1	0	0	0
0,3	0	0	-1	1	0	0	0	0	0
0	0,7	0	0	0	-1	1	0	0	0
0	0,3	0	0	-1	1	0	0	0	0
0	0	0,363122	0	0	0	0	0	0	0
0	0	0	0,0722697	0	0	0	-1	0	0
0	0	0	0	0,1676378	0	0	1	0	-1
0	0	0	0	0	0,1326589	0	0	-1	1
0	0	0	0	0	0	0,0311933	0	1	0
0	0	0	0	0	0	0	0	0	1

Gd	
0,7	0
0,3	0
0	0,7
0	0,3

MuM
805,1644
345,0704
1889,499
809,7853

iterations	Node 1	Node 2	Node 3	Node 4
0	803,88	344,52	1910,39	818,74
1	6594,33	2826,14	969,90	415,67
2	9583,04	4107,02	-4165,58	-1785,25
3	7015,83	3006,78	-2062,66	-884,00
4	805,16	345,07	1889,50	809,79

iter2

0,7	0,3	0	0	0	0	0	0	0	0
0	0	0,7	0,3	0	0	0	0	0	0
1	0	0	0	0,367273	0	0	0	0	0
1	-1	0	0	0	0,1149248	0	0	0	0
0	1	0	-1	0	0	0,13787	0	0	0
0	0	-1	1	0	0	0	0,09294	0	0
-1	0	1	0	0	0	0	0	0,051241	0
0	0	0	0	0	-1	1	0	0	0
0	0	0	0	0	0	0	-1	1	0
0	0	0	0	0	0	-1	1	0	1

$\cdot \begin{bmatrix} \Delta M^{(k)} \\ \Delta \mathbf{h}^{(k)} \end{bmatrix}$

inversa (jacobiano transp*jacobiano) | iter2

1,330724	0,228311	0,46893	-1,09417	-3,62325	-9,592472	-9,5925	16,81829	16,81829	-26,41076	1,09E-11		delta Mg	-3667,444
-0,33072	0,771689	0,53107	2,09417	0,900484	9,5924719	9,59247	-16,8183	-16,8183	26,410762	2,18E-11	1,1E-10		3004,168
5,66E-14	2,13E-13	1,58E-13	-9,3E-14	2,722767	1,407E-14	2,9E-14	3,07E-15	-1,5E-14	1,846E-14	3,46E-11			663,27567
0,217724	-0,508022	0,076723	-0,17902	-0,59281	2,3863777	2,38638	2,751692	2,751692	-0,365314	1,57E-11			-3066,152
-0,18149	0,423485	-0,06396	0,14923	0,494164	5,2641192	5,26412	-2,2938	-2,2938	7,5579147	243,6035			-1965,919
-0,08228	0,191978	-0,22328	0,52098	0,224019	2,3863777	2,38638	2,751692	2,751692	-0,365314	-312,2029		delta q_{uq}	-2867,17
0,149231	-0,348205	0,404974	-0,94494	-0,40632	-4,328353	-4,3284	14,52449	14,52449	-18,85285	-312,2087			-4970,087
0,025022	-0,058384	0,008817	-0,02057	-0,06813	-0,725746	0,27425	0,316238	0,316238	-0,041984	-168,6926			-40,17415
-0,00765	0,017843	-0,02075	0,04842	0,02082	0,2217911	0,22179	-0,74426	0,255744	0,9660475	-351,4581			-96,78353
-3,6E-15	1,78E-15	3,55E-15	3,2E-15	6,87E-15	2,771E-14	2,8E-14	-2,7E-14	-2,7E-14	1	1		delta H_{uH}	1

iter3

0,7	0,3	0	0	0	0	0	0	0	0
0	0	0,7	0,3	0	0	0	0	0	0
1	0	0	0	0,340468	0	0	0	0	0
1	-1	0	0	0	0,0688123	0	0	0	0
0	1	0	-1	0	0	0,22729	0	0	0
0	0	-1	1	0	0	0	0,226846	0	0
-1	0	1	0	0	0	0	0	0,217748	0
0	0	0	0	0	-1	1	0	0	0
0	0	0	0	0	0	0	-1	1	0
0	0	0	0	0	0	-1	1	0	1

$\cdot \begin{bmatrix} \Delta M^{(k)} \\ \Delta \mathbf{h}^{(k)} \end{bmatrix}$

inversa (jacobiano transp*jacobiano) | iter3

1,121647	0,716156	0,267081	-0,62319	-3,29443	-5,892713	-5,8927	3,924567	3,924567	-9,817279	9,09E-13		delta Mg	3005,4405
-0,12165	0,283844	0,732919	1,62319	0,357295	5,8927127	5,89271	-3,92457	-3,92457	9,8172792	4,55E-13	4,8E-11		-2445,465
2,44E-14	-9,66E-15	-1,1E-13	-8,3E-15	2,937137	-5,38E-15	-5E-15	1,53E-16	-1,8E-14	-2,33E-15	9,09E-13			-559,9754
0,258295	-0,602688	0,061504	-0,14351	-0,75865	2,0202445	2,02024	0,903755	0,903755	1,1164892	1,14E-12			2232,5266
-0,0782	0,182465	-0,01862	0,04345	0,229683	3,7880583	3,78806	-0,27361	-0,27361	4,061673	-190,6535			1330,8945
-0,04171	0,097312	-0,2385	0,55649	0,122494	2,0202445	2,02024	0,903755	0,903755	1,1164892	-237,5914		delta q_{uq}	2064,534
0,043448	-0,101378	0,248461	-0,57974	-0,12761	-2,104654	-2,1047	3,650952	3,650952	-5,755606	692,7138			3776,3596
0,017774	-0,041472	0,004232	-0,00988	-0,0522	-0,860982	0,13902	0,062189	0,062189	0,0768282	688,458			391,21669
-0,00946	0,022075	-0,0541	0,12624	0,027787	0,458284	0,45828	-0,79499	0,205013	1,2532709	603,1671			-219,1271
8,88E-16	-8,88E-16	0	1,1E-15	-1E-15	7,633E-16	2,2E-16	-2,2E-15	-2E-15	1	1		delta H_{uH}	1

iter4

0,7	0,3	0	0	0	0	0	0	0	0
0	0	0,7	0,3	0	0	0	0	0	0
1	0	0	0	0,363122	0	0	0	0	0
1	-1	0	0	0	0,0722697	0	0	0	0
0	1	0	-1	0	0	0,16764	0	0	0
0	0	-1	1	0	0	0	0,132659	0	0
-1	0	1	0	0	0	0	0	0,031193	0
0	0	0	0	0	-1	1	0	0	0
0	0	0	0	0	0	0	-1	1	0
0	0	0	0	0	0	-1	1	0	1

$\cdot \begin{bmatrix} \Delta M^{(k)} \\ \Delta \mathbf{h}^{(k)} \end{bmatrix}$

inversa (jacobiano transp*jacobiano) | iter4

1,263559	0,385028	0,708354	-1,65283	-3,47971	-12,15629	-12,156	17,79889	17,79889	-29,95518	-7030,244		delta Mg	8209,1826
-0,26356	0,614972	0,291646	2,65283	0,725815	12,156292	12,1563	-17,7989	-17,7989	29,955178	3563,346	6E-12		-4901,425
-1,6E-13	4,52E-14	-2,7E-13	-2,9E-13	2,753896	1,196E-14	1,8E-14	1,39E-14	-7,2E-15	2,228E-14	-5663,982			-14866,06
0,264878	-0,618048	0,148491	-0,34648	-0,72945	1,6199728	1,61997	3,731147	3,731147	-2,111174	-2427,421			-1553,992
-0,11419	0,266444	-0,06402	0,14937	0,314468	5,2668604	5,26686	-1,60852	-1,60852	6,8753798	-5398,191			-453,4004
-0,03512	0,081952	-0,15151	0,35352	0,096724	1,6199728	1,61997	3,731147	3,731147	-2,111174	-395,1873		delta q_{uq}	-1410,394
0,149369	-0,348528	0,644339	-1,50346	-0,41135	-6,889432	-6,8894	16,19037	16,19037	-23,0798	205,8737			-3643,379
0,019143	-0,044666	0,010731	-0,02504	-0,05272	-0,882925	0,11707	0,269649	0,269649	-0,152574	-0,892045			282,8808
-0,00466	0,010872	-0,0201	0,0469	0,012831	0,2149038	0,2149	-0,50503	0,49497	0,719934	-298,8581			-185,2092
3,55E-15	-2,66E-15	3,55E-15	0	-1,1E-15	4,913E-15	5E-15	3,22E-14	3,23E-14	1	1		delta H_{uH}	1

[23:00]

$$\mathbf{Aq} - \mathbf{M} = 0 \quad \text{and} \quad [\mathbf{I}_{kM} \quad \mathbf{I}_{uM}] \cdot \begin{bmatrix} \mathbf{M}_{kM} \\ \mathbf{M}_{uM} \end{bmatrix} - [\mathbf{A}_{kq} \quad \mathbf{A}_{uq}] \cdot \begin{bmatrix} \mathbf{q}_{kq} \\ \mathbf{q}_{uq} \end{bmatrix} = 0$$

A(uq)	1	2	3	4	5
1	-1	-1	0	0	1
2	0	1	-1	0	0
3	0	0	0	1	-1
4	0	0	1	-1	0

A(kq)	1	2	3	4	5
1	-1	-1	0	0	1
2	0	1	-1	0	0
3	0	0	0	1	-1
4	0	0	1	-1	0

I(uQ)			
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

M(uM)
0
0
0
0

Hazen-williams	R
e= 1,85	0,000181
Ku= 10,44	0,000181
	0,000181
	0,000181
	0,000181

q(kq)
26133
0
0
0
0

q(uq)
0
2823,63
-245,37
-1571,97
-2239,77

I(kQ)			
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

M(kM)	
818	1148,39927 G1
330,399	2729,13314 G2
1855,27	
873,865	

3877,53

I(uq)
1
1
1
1
1

I(uq)*eRq(Uq)^(e-1)				
0,3761	0	0	0	0
0	0,28722	0	0	0
0	0	0,036	0	0
0	0	0	0,174587	0
0	0	0	0	0,23589

eR|quq|^(e-1)				
0,376103	0	0	0	0
0	0,287224	0	0	0
0	0	0,036007	0	0
0	0	0	0,1745871	0
0	0	0	0	0,235889

H(uH)
17919,2
18072,11
17923,95

H(kH)
0
0
0

A10H0=	2300
	0
	0
	0
	0

Gd	
0,7	0
0,3	0
0	0,7
0	0,3

Mg
1148,4
2729,1

A(uH)									
0	-1	1	0	0	-1	0	0	1	1
0	0	0	-1	1	1	-1	1	0	0
0	0	-1	1	0	1	1	0	0	1

A(kH)									

A^T(uH)			
0	0	0	
-1	0	0	
1	0	-1	
0	-1	1	
0	1	0	

A^T(kH)			
0	0	0	
-1	0	0	
1	0	-1	
0	-1	1	
0	1	0	

h(kh)
602,0117
54,95648
-23,1559
-200,438

I(kh)				
1	0	0	0	
0	1	0	0	
0	0	1	0	
0	0	0	1	
0	0	0	0	

I(uh)				
1	0	0	0	
0	1	0	0	
0	0	1	0	
0	0	0	1	
0	0	0	0	

jacobiano(nxm)

index	| quq | Hkh	1	2	1	2	3	4	5	2	3	4
1	1148,4	0,7	0	1	1	0	0	-1	0	0	0
2	2729,13	0,3	0	0	-1	1	0	0	0	0	0
1	3877,53	0	0,7	0	0	0	-1	1	0	0	0
2	2823,63	0	0,3	0	0	-1	1	0	0	0	0
3	-245,37	0	0	0,3761	0	0	0	0	0	0	0
4	-1571,97	0	0	0	0,287224	0	0	0	-1	0	0
5	-2239,77	0	0	0	0	0,03601	0	0	1	0	-1
2	17919,2	0	0	0	0	0	0,174587	0	0	-1	1
3	18072,1	0	0	0	0	0	0	0,235889	0	1	0
4	17924	0	0	0	0	0	0	0	0	0	1

jacobiano transposto

0,7	0,3	0	0	0	0	0	0	0	0
0	0	0,7	0,3	0	0	0	0	0	0
1	0	0	0	0,3761	0	0	0	0	0
1	-1	0	0	0	0,287224	0	0	0	0
0	1	0	-1	0	0	0,03601	0	0	0
0	0	-1	1	0	0	0	0,17458712	0	0
-1	0	1	0	0	0	0	0	0,235889	0
0	0	0	0	0	-1	1	0	0	0
0	0	0	0	0	0	0	-1	1	0
0	0	0	0	0	0	-1	1	0	1

MuM
803,879
344,52
1910,39
818,74

$$\cdot \begin{bmatrix} \Delta \mathbf{M}^{(k)} \\ \Delta \mathbf{h}^{(k)} \end{bmatrix}$$

inversa (jacobiano transp*jacobiano)*transposto

1,61186	-0,4277	0,2929	-0,68335	-4,2857	-7,10085	-7,10085	5,591595	5,591595	-12,69245
-0,6119	1,42768	0,7071	1,683354	1,62685	7,100853	7,100853	-5,5916	-5,5916	12,692448
1,3E-13	-2E-13	-8E-14	1,37E-13	2,65885	1,77E-14	-9,4E-16	-5,3E-15	-2,3E-14	1,473E-14
0,05387	-0,1257	0,0098	-0,02284	-0,1432	2,856461	2,856461	0,186865	0,186865	2,6695963
-0,4297	1,00261	-0,078	0,182169	1,14248	4,986717	4,986717	-1,49061	-1,49061	6,4773308
-0,2461	0,57431	-0,29	0,677163	0,65443	2,856461	2,856461	0,186865	0,186865	2,6695963
0,18217	-0,4251	0,2148	-0,50118	-0,4844	-2,11414	-2,11414	4,100981	4,100981	-6,215117
0,01547	-0,0361	0,0028	-0,00656	-0,0411	-0,17956	0,820445	0,053672	0,053672	0,766773
-0,043	0,10027	-0,051	0,118224	0,11426	0,498701	0,498701	-0,96738	0,032624	1,4660771
8,9E-16	3,6E-15	3E-15	2,22E-15	3,9E-15	1,57E-14	1,55E-14	-5,3E-15	-5,3E-15	1

*

-10563	delta Mg	18912,42
2394,08		-18951,5
-3097,9		-11593,5
-366,01		-20069,6
-4360,4		-23349,2
-3713	delta quq	-18029,8
-2893,2		-7861,62
-2627,6		-2051,45
-2373,7		-519,205
1	delta HuH	1

iter1	
index	g | q_{uq} | H_{uH}
1	20060,82
2	-16222,3
1	-7716
2	-17246
3	-23594,6
4	-19601,8
5	-10101,4
2	15867,75
3	17552,91
4	17924,95

MuM
14042,57
6018,246
-11355,6
-4866,7

R|quq|^(e-a)					I(uq)	I(uq)*Rq(Uq)				
0,36488	0	0	0	0	1		0,364876955	0	0	0	0
0	0,722848	0	0	0	1		0	0,722848	0	0	0
0	0	0,943526	0	0	1		0	0	0,943526	0	0
0	0	0	0,805961	0	1		0	0	0	0,805961	0
0	0	0	0	0,458761521	1		0	0	0	0	0,458761521

0,7	0	1	1	0	0	-1	0	0	0
0,3	0	0	-1	1	0	0	0	0	0
0	0,7	0	0	0	-1	1	0	0	0
0	0,3	0	0	-1	1	0	0	0	0
0	0	0,364877	0	0	0	0	0	0	0
0	0	0	0,722848	0	0	0	-1	0	0
0	0	0	0	0,943525686	0	0	1	0	-1
0	0	0	0	0	0,805960673	0	0	-1	1
0	0	0	0	0	0	0,458761521	0	1	0
0	0	0	0	0	0	0	0	0	1

0,7	0,3	0	0	0	0	0	0	0	0
0	0	0,7	0,3	0	0	0	0	0	0
1	0	0	0	0,364876955	0	0	0	0	0
1	-1	0	0	0	0,72284838	0	0	0	0
0	1	0	-1	0	0	0,943525686	0	0	0
0	0	-1	1	0	0	0	0,805960673	0	0
-1	0	1	0	0	0	0	0	0,458762	0
0	0	0	0	0	-1	1	0	0	0
0	0	0	0	0	0	0	-1	1	0
0	0	0	0	0	0	-1	1	0	1

inversa (jacobiano transp*jacobiano)									
1,30708	0,283474	0,451127	-1,05263	-3,582255747	-1,416076447	-1,416076447	1,865795572	1,865796	-3,28187
-0,30708	0,716526	0,548873	2,05263	0,841605822	1,416076447	1,416076447	-1,865795572	-1,865796	3,281872
-4E-15	-8,9E-16	-7,1E-15	-8,1E-15	2,740649925	-2,66454E-15	2,96059E-16	4,08932E-15	1,11E-15	4,14E-15
0,22203	-0,51806	0,07663	-0,1788	-0,608497884	0,359564315	0,359564315	0,316932329	0,316932	0,042632
-0,1701	0,396895	-0,05871	0,136985	0,46617884	0,784387249	0,784387249	-0,242806342	-0,242806	1,027194
-0,07797	0,181937	-0,22337	0,521196	0,213697093	0,359564315	0,359564315	0,316932329	0,316932	0,042632
0,13698	-0,31963	0,39242	-0,91565	-0,375426982	-0,631689198	-0,631689198	1,62298923	1,622989	-2,25468
0,16049	-0,37448	0,055392	-0,12925	-0,43985171	-0,740089517	0,259910483	0,229094021	0,229094	0,030816
-0,06284	0,146634	-0,18003	0,420063	0,172231453	0,289794697	0,289794697	-0,744565007	0,255435	1,03436
-8,9E-16	0	0	-5,2E-16	-3,32367E-17	3,46945E-18	6,66134E-16	2,22045E-16	-2,22E-16	1

$$\begin{bmatrix} \Delta \mathbf{M}^{(k)} \\ \Delta \mathbf{h}^{(k)} \end{bmatrix}$$

-3,3E-10	12956,89	delta Mg	-11289,8
3,3E-11	3463,866		11527,18
-1,1E-10			-237,394
2,7E-10			18200,99
-86,6195			21587,92
11864,2		delta q_{uq}	18129,77
21660,1			10060,74
15196,2			1292,349
4032,12			-583,363
1		delta H_{uH}	1

iter2

index	$\lvert q_{uq} \rvert H_{uH}$
1	8771,04
2	-4695,17
1	-7953,4
2	955,023
3	-2006,69
4	-1472
5	-40,6499
2	17160,1
3	16969,5
4	17925,9

MuM
6139,72
2631,31
-3286,62
-1408,55

R|quq|^(e-a)					I(uq)	I(uq)*Rq(Uq)				
0,3744	0	0	0	0	1	0,3743973	0	0	0	0
0	0,061784	0	0	0	1	0	0,061784	0	0	0
0	0	0,11614	0	0	1	0	0	0,11614	0	0
0	0	0	0,089245	0	1	0	0	0	0,08925	0
0	0	0	0	0,004222	1	0	0	0	0	0,004222

0,7	0	1	1	0	0	-1	0	0	0
0,3	0	0	-1	1	0	0	0	0	0
0	0,7	0	0	0	-1	1	0	0	0
0	0,3	0	0	-1	1	0	0	0	0
0	0	0,3744	0	0	0	0	0	0	0
0	0	0	0,061784	0	0	0	-1	0	0
0	0	0	0	0,116137	0	0	1	0	-1
0	0	0	0	0	0,089245	0	0	-1	1
0	0	0	0	0	0	0,004222	0	1	0
0	0	0	0	0	0	0	0	0	1

iter3

index	$\lvert q_{uq} \rvert H_{uH}$
1	-12291
2	16164,6
1	-7751,16
2	-9931,08
3	-6574,19
4	-12297,4
5	-25467,9
2	17089,5
3	16475,1
4	17926,9

MuM
-8603,69
-3687,3
11315,2
4849,38

R|quq|^(e-a)					I(uq)	I(uq)*Rq(Uq)				
0,3663	0	0	0	0	1	0,3662894	0	0	0	0
0	0,452179	0	0	0	1	0	0,452179	0	0	0
0	0	0,31844	0	0	1	0	0	0,31844	0	0
0	0	0	0,542257	0	1	0	0	0	0,54226	0
0	0	0	0	1,006833	1	0	0	0	0	1,006833

0,7	0	1	1	0	0	-1	0	0	0
0,3	0	0	-1	1	0	0	0	0	0
0	0,7	0	0	0	-1	1	0	0	0
0	0,3	0	0	-1	1	0	0	0	0
0	0	0,36629	0	0	0	0	0	0	0
0	0	0	0,452179	0	0	0	-1	0	0
0	0	0	0	0,318441	0	0	1	0	-1
0	0	0	0	0	0,542257	0	0	-1	1
0	0	0	0	0	0	1,006833	0	1	0
0	0	0	0	0	0	0	0	0	1

iter4

index	$\lvert q_{uq} \rvert H_{uH}$
1	10555,2
2	-6510,02
1	-7922,73
2	-116,072
3	-3613,04
4	-2533,9
5	167,852
2	17639,1
3	15704,1
4	17927,9

R|quq|^(e-a)					I(uq)	I(uq)*Rq(Uq)				
0,3732	0	0	0	0	1	0,3731697	0	0	0	0
0	0,010301	0	0	0	1	0	0,010301	0	0	0
0	0	0,19145	0	0	1	0	0	0,19145	0	0
0	0	0	0,141607	0	1	0	0	0	0,14161	0
0	0	0	0	0,014095	1	0	0	0	0	0,014095

0,7	0	1	1	0	0	-1	0	0	0
0,3	0	0	-1	1	0	0	0	0	0
0	0,7	0	0	0	-1	1	0	0	0
0	0,3	0	0	-1	1	0	0	0	0
0	0	0,37317	0	0	0	0	0	0	0
0	0	0	0,010301	0	0	0	-1	0	0
0	0	0	0	0,19145	0	0	1	0	-1
0	0	0	0	0	0,141607	0	0	-1	1
0	0	0	0	0	0	0,014095	0	1	0
0	0	0	0	0	0	0	0	0	1

Gd	
0,7	0
0,3	0
0	0,7
0	0,3

MuM
7388,7
3166,6
-4557
-1953

iteration	Node 1	Node 2	Node 3	Node 4
0	803,8795	344,5198	1910,393	818,7399
1	14042,57	6018,246	-11355,6	-4866,7
2	6139,725	2631,311	3286,619	1408,551
3	-8603,69	-3687,3	11315,23	4849,384
4	7388,653	3166,566	-4557,02	-1953,01

H1(ft)	H0(ft)	q0(gpm)	Δ t(min)	A(ft^2)
26039	2529	7922,73	60	5808,805

iter2

0,7	0,3	0	0	0	0	0	0	0	0
0	0	0,7	0,3	0	0	0	0	0	0
1	0	0	0	0,374397	0	0	0	0	0
1	-1	0	0	0	0,061784	0	0	0	0
0	1	0	-1	0	0	0,116136964	0	0	0
0	0	-1	1	0	0	0	0,08924516	0	0
-1	0	1	0	0	0	0	0	0,00422244	0
0	0	0	0	0	-1	1	0	0	0
0	0	0	0	0	0	0	-1	1	0
0	0	0	0	0	0	-1	1	0	1

$$\cdot \begin{bmatrix} \Delta \mathbf{M}^{(k)} \\ \Delta \mathbf{h}^{(k)} \end{bmatrix}$$

inversa (jacobiano transp*jacobiano)				iter2								
1,458026	-0,068728	1,25941	-2,93861	-3,89433	-24,7112	-24,71122238	47,03918729	47,03918729	-71,75040967	-5,5E-12	3,57E-10	delta M_g	-21062
-0,458026	1,068728	-0,25941	3,938614	1,223369	24,71122	24,71122238	-47,03918729	-47,03918729	71,75040967	1,36E-12	3,96E-10		20859,78
1,55E-12	2,18E-12	1,1E-12	-9,9E-13	2,67096	1,75E-15	-2,7441E-14	-4,29766E-14	-2,44423E-15	1,56726E-14	1,86E-11			202,2417
0,285516	-0,666204	0,24662	-0,57545	-0,7626	0,781438	0,781438228	9,211383467	9,211383467	-8,429945239	1,14E-12			-10886,1
-0,151892	0,354414	-0,1312	0,306134	0,405697	8,194805	8,194804941	-4,900372721	-4,900372721	13,09517766	75,71874			-4567,5
-0,014484	0,033796	-0,05338	0,12455	0,038686	0,781438	0,781438228	9,211383467	9,211383467	-8,429945239	-602,015		delta q_{uq}	-10825,4
0,306134	-0,714313	1,12821	-2,63248	-0,81767	-16,5164	-16,51641744	42,13881457	42,13881457	-58,65523201	-602,025			-25427,3
0,01764	-0,041161	0,01524	-0,03555	-0,04712	-0,95172	0,048280235	0,56911441	0,56911441	-0,520834175	-470,643			-70,5697
-0,001293	0,003016	-0,00476	0,011115	0,003453	0,06974	0,06973958	-0,177928611	0,822071389	0,24766819	-601,84			-494,475
0	-8,88E-15	0	1,24E-14	-4,2E-15	7,08E-14	6,9722E-14	-5,72875E-14	-5,77316E-14	1	1		delta H_{uH}	1

iter3

0,7	0,3	0	0	0	0	0	0	0	0
0	0	0,7	0,3	0	0	0	0	0	0
1	0	0	0	0,366289	0	0	0	0	0
1	-1	0	0	0	0,452179	0	0	0	0
0	1	0	-1	0	0	0,318440654	0	0	0
0	0	-1	1	0	0	0	0,542257027	0	0
-1	0	1	0	0	0	0	0	1,006833077	0
0	0	0	0	0	-1	1	0	0	0
0	0	0	0	0	0	0	-1	1	0
0	0	0	0	0	0	-1	1	0	1

$$\cdot \begin{bmatrix} \Delta \mathbf{M}^{(k)} \\ \Delta \mathbf{h}^{(k)} \end{bmatrix}$$

inversa (jacobiano transp*jacobiano)				iter3								
1,304062	0,290522	0,18139	-0,42325	-3,5602	-2,24146	-2,241460674	1,115049835	1,115049835	-3,356510509	1,27E-11	2,82E-11	delta M_g	22846,21
-0,304062	0,709478	0,81861	1,423251	0,830114	2,241461	2,241460674	-1,115049835	-1,115049835	3,356510509	1,14E-11	3,57E-11		-22674,6
-3,25E-14	-1,09E-14	-2,1E-14	0	2,730082	1,67E-15	3,62673E-15	7,91959E-15	-4,88498E-15	3,36768E-15	-1,3E-11			-171,572
0,161662	-0,377212	0,02249	-0,05247	-0,44135	1,019788	1,019788145	0,138230604	0,138230604	0,881557541	5,46E-12			9815,011
-0,229557	0,535632	-0,03193	0,074506	0,626708	1,692226	1,692226347	-0,196284346	-0,196284346	1,888510693	-62,8451			2961,148
-0,138338	0,322788	-0,27751	0,64753	0,377674	1,019788	1,019788145	0,138230604	0,138230604	0,881557541	3888,611		delta q_{uq}	9763,539
0,074506	-0,173846	0,14946	-0,34874	-0,20341	-0,54923	-0,549234327	0,918765489	0,918765489	-1,467999816	1491,476			25635,78
0,0731	-0,170567	0,01017	-0,02373	-0,19957	-0,53887	0,461126336	0,062504916	0,062504916	0,39862142	6066,359			549,5263
-0,075015	0,175034	-0,15048	0,351128	0,204796	0,552987	0,552987287	-0,925043484	0,074956516	1,478030771	25039,94			-771,011
-4,44E-16	2,22E-16	-4,4E-16	1,11E-16	8,53E-16	1,33E-15	1,55431E-15	-1,9984E-15	-1,33227E-15	1	1		delta H_{uH}	1

iter4

0,7	0,3	0	0	0	0	0	0	0	0
0	0	0,7	0,3	0	0	0	0	0	0
1	0	0	0	0,37317	0	0	0	0	0
1	-1	0	0	0	0,010301	0	0	0	0
0	1	0	-1	0	0	0,19144958	0	0	0
0	0	-1	1	0	0	0	0,141606517	0	0
-1	0	1	0	0	0	0	0	0,014094549	0
0	0	0	0	0	-1	1	0	0	0
0	0	0	0	0	0	0	-1	1	0
0	0	0	0	0	0	-1	1	0	1

$$\cdot \begin{bmatrix} \Delta \mathbf{M}^{(k)} \\ \Delta \mathbf{h}^{(k)} \end{bmatrix}$$

inversa (jacobiano transp*jacobiano)				iter4								
1,04401	0,897311	0,78392	-1,82915	-2,79768	-14,2412	-14,24115066	18,45306109	18,45306109	-32,69421176	-3,5E-11	4,87E-11	delta M_g	-8546,67
-0,04401	0,102689	0,21608	2,829152	0,117935	14,24115	14,24115066	-18,45306109	-18,45306109	32,69421176	1,27E-11	9,55E-12		8400,599
-7,85E-13	-7,3E-13	-3,3E-13	1,78E-13	2,679746	-2E-15	-2,48632E-14	-1,21085E-14	-1,12172E-14	8,67824E-15	-3,6E-12			146,0754
0,297211	-0,693493	0,22317	-0,52073	-0,79645	0,902408	0,902407564	5,253263287	5,253263287	-4,350855723	5E-12			-4961,52
-0,015992	0,037314	-0,01201	0,028018	0,042854	5,174753	5,174752763	-0,282655041	-0,282655041	5,457407804	54,51091			-2397,52
-0,002789	0,006507	-0,07683	0,179273	0,007473	0,902408	0,902407564	5,253263287	5,253263287	-4,350855723	-600,816		delta q_{uq}	-4917,7
0,028018	-0,065375	0,77191	-1,80113	-0,07508	-9,0664	-9,066397901	18,17040605	18,17040605	-27,23680395	89,7028			-10798,1
0,003062	-0,007144	0,0023	-0,00536	-0,0082	-0,9907	0,009295756	0,054114189	0,054114189	-0,044818433	-243,196			549,707
-0,000395	0,000921	-0,01088	0,025386	0,001058	0,127787	0,127786792	-0,256103683	0,743896317	0,383890475	-604,377			-452,183
0	-2,66E-15	-7,1E-15	6,22E-15	-3,8E-15	2,26E-14	2,17604E-14	-1,35447E-14	-1,35447E-14	1	1		delta H_{uH}	1